Combinatoric Models of Information Retrieval Ranking Methods and Performance Measures for Weakly-Ordered Document Collections

Lewis Church

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the School of Information and Library Science.

Chapel Hill
2010

Approved by:

Robert M. Losee, Advisor

Robert E. Burgin, Committee Member

Claudia J. Gollop, Committee Member

Jane Greenberg, Committee Member

Richard Marciano, Committee Member

Paul S. Solomon, Committee Member

Abstract

LEWIS CHURCH: Combinatoric Models of Information Retrieval Ranking Methods and Performance Measures for Weakly-Ordered Document Collections
(Under the direction of Robert M. Losee)

This dissertation answers three research questions: (1) What are the characteristics of a combinatoric measure, based on the Average Search Length (ASL), that performs the same as a probabilistic version of the ASL?; (2) Does the combinatoric ASL measure produce the same performance result as the one that is obtained by ranking a collection of documents and calculating the ASL by empirical means?; and (3) When does the ASL and either the Expected Search Length, MZ-based E, or Mean Reciprocal Rank measure both imply that one document ranking is better than another document ranking?

Concepts and techniques from enumerative combinatorics and other branches of mathematics were used in this research to develop combinatoric models and equations for several information retrieval ranking methods and performance measures. Empirical, statistical, and simulation means were used to validate these models and equations.

The document cut-off performance measure equation variants that were developed in this dissertation can be used for performance prediction and to help study any vector V of ranked documents, at arbitrary document cut-off points, provided that (1) relevance is binary and (2) the following information can be determined from the ranked output: the document equivalence classes and their relative sequence, the number of documents in each equivalence class, and the number of relevant documents that each class contains. The performance measure equations yielded correct values for both strongly- and weakly-ordered document collections.

Dedication

To my mother, Arlen Elizabeth Church

To the memory of my father, Lewis Church, Sr.

To my wife, Dr. Lila Teresa Church

To all the other wonderful people, far too numerous to mention, who provided encouragement, inspiration, and believed in me during my academic journey

Acknowledgments

To the casual observer, a doctoral dissertation may appear to be solitary work. Completing a project of this magnitude requires a network of support, however, and I am indebted to many people. I am especially grateful to my dissertation advisor and committee chair, Dr. Robert M. Losee, and his fellow committee members Dr. Robert E. Burgin, Dr. Claudia J. Gollop, Dr. Jane Greenberg, Dr. Richard Marciano, and Dr. Paul Solomon.

The original members of my committee remained with me even though this journey took more time than I expected to complete the dissertation. I especially thank Dr. Richard Marciano for agreeing to come on as a new committee member at a point where my dissertation work was at a very advanced stage.

I am indebted to my wife, Lila Teresa Church, who was a doctoral student for most of the years that I was a doctoral student. She provided much advice and encouragement during these years. It was wonderful not having to explain to her why I had to study so much and how my life was continuously being impacted by the many demands that the doctoral program placed upon me.

I would be remiss if I did not thank Robert Ray, my manager at SAS Institute Inc. in Cary, North Carolina, for allowing me the flexibility to work full-time at SAS, take classes (during business hours) at a university that was about 25 miles away, and to make up the lost time by working longer hours some days after class. Robert, and my other departmental colleagues at SAS, also provided me with various kinds of encouragement during the years. Thanks very much! I really appreciated that.

There were many other people that provided encouragement to me over the years. Even if I tried to list them here, I would probably miss more than a few of their names. So, in lieu of trying to enumerate them, I just want to say, to all of them, that the encouragement and other acts of kindness, that you sent my way, were very much appreciated. Thanks!

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ASK	Anomalous State of Knowledge
ASL	Average Search Length
CF	Cystic Fibrosis
CLM	coordination level matching
DB	database system
DT	decision-theoretic
ESL	Expected Search Length
GCD	greatest common divisor
IDF	inverse document frequency
IR	information retrieval
IRS	information retrieval system
IRDB	information retrieval-database

IRDBS	information retrieval-database system
MRR	Mean Reciprocal Rank
MZE	MZ-based E measure
ROC	Receiver Operating Characteristics

List of Symbols and Notation

$[k]$	the set of the first k positive integers	22
N	number of documents in a collection	24
R	number of relevant documents in a collection	24
S	number of non-relevant documents in a collection	24
n_i	number of documents with feature frequency i	24
r_i	number of relevant documents with feature frequency i	24
s_i	number of non-relevant documents with feature frequency i	24
$\binom{n}{k}$	number of combinations of n things taken k at a time	26
$C_k(n)$	number of compositions of n into k parts	26
$\tilde{C}_k(n)$	number of weak compositions of n into k parts	26
$C(n,k)$	number of combinations of n things taken k at a time	26
$P(n,k)$	number of permutations of n things taken k at a time	26

$n!$	number of permutations of n distinct objects	26
Σ	summation	51
$\log a$	natural logarithm of a	51
$P(a)$	probability of event a	51
$P(a\|b)$	probability of event a given that event b has occurred	51
$\Pr(a)$	probability of event a	51
$\Pr(a\|b)$	probability of event a given that event b has occurred	51
Π	multiplication	52
$a \propto b$	a is proportional to b	54
$\lvert a\rvert$	cardinality of set/bag a	66
$\lfloor a \rfloor$	the greatest integer that is less than or equal to a	83
$\lceil a \rceil$	the least integer that is greater than or equal to a	83
$\max(a, b)$	maximum of a and b	90
$\min(a, b)$	minimum of a and b	90

$\mathbb{N}$	the set of natural numbers	94
$\mathcal{Q}$	quality of a ranking method	113
$\tau(N)$	the number of unique divisors of a positive integer N	151
$\mathbb{Z}^+$	the set of positive integers	156
$\mathcal{A}$	normalized average position of a relevant document	262
ASL	Average Search Length	263
$@k$	document cut-off at position k in ranked vector V of documents	449
ESL$@k(V,x)$	ESL at position k for a request of x documents	449
ASL$@k(V)$	ASL at position k in a ranked vector V of documents	459
P$@k(V)$	precision at position k in a ranked vector V of documents	460
R$@k(V)$	recall at position k in a ranked vector V of documents	461
MZE$@k(V)$	MZE at position k in a ranked vector V of documents	462
RR$@k(V)$	reciprocal rank at position k in a ranked vector V of documents	463
$a \leftarrow b$	assign the value denoted by b to a	515

This page intentionally left blank

Chapter 1

Introduction

The purpose of this research was to investigate the characteristics of analytic measures for studying and predicting the performance of information retrieval (IR) systems and of systems that had both information retrieval and database capabilities. The use of these measures for *prediction*, rather than mainly for retrospection, was quite different from how many IR performance measures had been used in the past and were currently being used. Some related work, in a distributed database context, focuses on "analytical techniques for predicting the performance of various collection fusion scenarios" (Losee and Church, 2004).

Each of these types of systems was assumed to reference documents that were stored in a centralized database. In this dissertation, the former type of system was referred to as an information retrieval system (IRS) and the latter type was referred to as an information retrieval-database system (IRDBS). In particular, the research concentration in this dissertation was on a measure known as the Average Search Length (ASL) (Losee, 1998) and two measures that were closely related to it: the normalized average position of a relevant document ($\mathcal{A}$) and the quality of a ranking method ($\mathcal{Q}$).

This research had four main contributions: (1) combinatoric models for several quality of ranking measures; (2) combinatoric-based equations for the Average Search Length (Losee, 1998), the Expected Search Length (Cooper, 1968), the MZ-based E measure (van

Rijsbergen, 1979), and the reciprocal rank measure (Voorhees, 2001) that were defined at all points in a ranking and yielded correct results for both strongly- and weakly-ordered document collections; (3) a method that generated graphs which illustrated regions of agreement and disagreement between two performance measures for a vector of ranked documents; and (4) a procedure that determined when two performance measures considered a document ranking for a vector V_1 to be better than a document ranking for a vector V_2.

The measures that were developed for this dissertation could aid in the understanding and prediction of system performance for single information retrieval queries that were submitted to either an information retrieval system or to an integrated information retrieval-database system. The mathematical models constructed for this research produced analytic results that were empirically validated.

There is a multitude of information retrieval performance measures, some more intuitive and easier to understand than others. Generally, a performance measure can be used in either a predictive or retrospective manner. Many of these measures can be more easily used retrospectively than predictively due to the problem of parameter estimation. Each performance measure possesses both strong and weak points. The determination of which measure is (more) appropriate for a particular situation is influenced by the goal(s) of the study or the audience.

A particularly appealing measure, one that can be used for either predictive, or retrospective, purposes is the Average Search Length. It was the main measure of interest in this dissertation. Some of its major strengths are that it is intuitive, easy to explain, and relatively straightforward to calculate. Another strong point is that it is a single number measure of performance and that it can be used to characterize the performance of systems that use ranking functions based on such diverse techniques as the inverse document frequency (Sparck Jones, 1972; Robertson, 1974; Robertson and Jones, 1976;

Croft and Harper, 1979), decision-theoretic ranking (Losee, 1998), and coordination level matching (Losee, 1987).

1.1 Research Goals and Questions

This research developed combinatorial equations for the Average Search Length (ASL) measure and its independent variables, namely, the normalized average position of a relevant document ($\mathcal{A}$), and the quality of a ranking method ($\mathcal{Q}$) in a centralized information retrieval context. This research also extended the ASL, MZ-based E measure (MZE) (van Rijsbergen, 1979), Expected Search Length (ESL) (Cooper, 1968), and Mean Reciprocal Rank (MRR) (Voorhees, 2001) measures in two ways: (1) the values of each of these measures were calculable at an arbitrary position in a ranking and (2) the calculated values were correct even if the documents in a ranking are weakly-ordered.

Combinatoric arguments were utilized to help develop this descriptive information and proofs were constructed to show the equivalence of these combinatoric-based equations with their respective probabilistic counterparts. These entities were used to help characterize and predict the performance of various scenarios when optimal ranking (Losee, 1998), worst-case ranking, random ranking, and various other degrees of non-optimal ranking were assumed. Definitions of the major concepts, that were introduced above, immediately follow the statement of the three research questions below.

The research questions were:

1. What would be the characteristics of a combinatoric measure (CM_ASL), based on the ASL, that performs the same as a probabilistic measure of retrieval performance, also based on the ASL?

2. Does the CM_ASL measure produce the same performance results as that of an actual document ranking? [In other words, is there any statistically significant difference between the predicted performance and the performances observed in

actual rankings?]

3. When does the ASL measure and one of these measures (i.e., MZE (van Rijsbergen, 1979), ESL (Cooper, 1968), and MRR (Voorhees, 2001)) both imply that one document ranking is better than another document ranking?

These three Research Questions (RQs) are occasionally referred to as RQ #1, RQ #2, and RQ #3, respectively, in the remainder of this dissertation.

1.2 Significance of this Research

The equations and techniques that were developed from this research could be used to predict and study performance, in terms of the Average Search Length (an intuitive measure for the user), for the inverse document frequency, coordination level matching, and decision-theoretic ranking methods ranking methods — without the need to estimate quality of ranking values from historical data. These equations and techniques could also be used to determine when two performance evaluation measures consider one document ranking to be better than another document ranking.

Based on the literature review for this dissertation, a novel aspect of this research was that these equations were developed by using a largely combinatoric approach. Combinatoric techniques and results from combinatorics opened up new avenues of exploration and provided more insight into how various parameters interacted to affect the inverse document frequency, coordination level matching, and decision-theoretic ranking methods.

Another significant feature was that this research extended the work in Losee (2000) to compare the performance between measures that are "based on the totality of the search process" (Losee, 2000) (e.g., ASL) as well as those that "determine performance at a point in the search process" (Losee, 2000) (e.g., MZE, ESL). This provided a way to compare more measures with respect to how well they agreed and disagreed with specific document rankings. The knowledge gained can help researchers and practitioners better understand the strengths and weaknesses of various ranking methods.

From an IR perspective, the use of combinatoric techniques means that the typical IR assumptions of term independence, uniformly distributed values, and equiprobable events can often be relaxed, or even eliminated, on a joint or individual basis, in order to develop better and more accurate models. Sometimes, during performance evaluation, if the probability of an event is not known, then this probability is either estimated or a subjective probability is provided. Combinatoric techniques often give researchers the ability to calculate an exact probability which can be used in lieu of an estimated or subjective one.

In particular, several combinatoric concepts and techniques are used in upcoming chapters to enable the calculation of exact (or close to exact) values for quality of ranking measures; normalized and unnormalized search lengths; and their associated means and standard deviations. Each of these are discussed in detail later, starting with the next chapter. Also, these chapters contain several illustrative examples. The concepts and techniques that are alluded to at the beginning of this paragraph include, but are not limited to, probability generating functions, Gaussian polynomials, compositions, partitions, k-subsets, permutations, combinations, asymptotics, and the Principle of Inclusion and Exclusion. Citations for these concepts and techniques are provided as they are introduced in the subsequent chapters.

Historically, the preponderance of performance research in information retrieval (IR) has been of an experimental nature concerned with *effectiveness* rather than *efficiency* (Vogt, 1999; Grossman and Frieder, 2004).

> IR effectiveness deals with retrieving the most relevant information to a user need, while IR efficiency deals with providing fast and ordered access to large amounts of information. IR efficiency ensures that systems scale up to the vast amounts of information available for retrieval, and is of utmost importance to both academic

and corporate environments. (Blanco and Silvestri, 2008)

Analytic models (Losee, 1998; Losee and Paris, 1999), where the focus is on prediction rather than experimentation, do not have the quantity of associated research as does research based on experimentation. In recent years, though, the interest in analytical research has increased. A large factor in this has been the ever-increasing size of document collections and the influence of the World Wide Web (Dong and Watters, 2004). During this time, there has been significant increases in computational speed (e.g., processor speed, memory access) and storage capacities with, of course, positive impacts on the performance of IR systems. However, the gains made in computational speed (which was growing at a linear rate) were more than offset by the growth of the sizes of document collections (which were growing at an exponential rate). “While people enjoy having access to this diverse data, they also have to face the problem of efficiently finding the information they really want” (Dong and Watters, 2004). This is a burden that should fall on the system, and not on the user.

A key to alleviating, or ameliorating this burden, is a better understanding of the search process and the impact that it has on the internal workings of a search engine and some of the choices that the engine has to make. Given the high degree of interest in the database and IR research communities in the development of IR systems that are either built on top of relational database systems or that integrate relational database and IR capabilities (e.g., IRDB systems), it is crucial that system developers have a better understanding of the factors that influence ranking, selectivity, and various execution costs. Analytic models of performance can help provide this insight. In addition, they can be used to help predict ranking, selectivity, and the choosing of one access plan over another in an IRDB system.

The primary motivation for the research and industrial interest in information retrieval-database (IRDB) systems has to do with the nature of today’s applications. “Modern applications such as customer support, health care, and digital libraries require capabilities for both data and text management" (Chaudhuri et al., 2005). Neither traditional database (DB) management systems nor IR systems are flexible enough to handle these types of applications because they require that these systems handle both structured data and text well. DB systems are very good at handling structured data such as customer records for a business whereas IR systems are good at handling unstructured entities such as text documents. Neither is much good at handling each other's bread-and-butter kinds of applications. Many years ago, when application uses did not have the degree of overlap that we have today, one could very much just exclusively use an IR or DB system, depending on the application. However, application requirements have changed much over just the past decade or so. The passage below provides insight into why systems that combine both IR and DB functionality are very important today.

> DB and IR systems are currently separate technologies. Thirty years ago, the application classes that drove the progress of these systems were disjoint and did indeed pose very different requirements: classical business applications like payroll or inventory management on the DB side, and abstracts of publications or patents on the IR side. The situation is radically different today. Virtually all advanced applications need both structured data and text documents, and information fusion is a central issue. Seamless integration of structured data and text is at the top of the wish lists of many enterprises. Example applications that would benefit include the following:
>
> - Customer support systems that track complaints and response threads and must ideally be able to automatically identify similar earlier requests.
> - Health care systems that have access to the electronic information produced by all hospitals, medical labs, and physicians (with appropriate measures for privacy and authorization), and have the capability of monitoring trends and generating early alerts about health crises such as epidemic diseases.
> - Intranet search engines for organizations with richly structured data sources in addition to traditional document repositories. (Chaudhuri et al., 2005)

One key difference between an IR and a DB system is that, normally, an IR system cannot be simply satisfied with retrieving the results for a query, but also has to order those results into a sequence. The results that are more likely to be relevant to a user are nearer the front of the sequence than those that are less likely to be relevant. This process is known as *ranking*. The primary benefit of ranking is that it puts the results (e.g., documents) into a known order and thereby saves the user from possibly having to inspect all of the documents just to find a few useful ones. There are many ways to rank documents. For example, the vector space model does it one way and the probabilistic model does it another way (Dominich, 2001). And, within the framework of a particular model, there are often several variations on that model's basic ranking algorithm. For example, the information retrieval literature has a variety of term weighting schemes that have been considered for the vector space model (Salton and Buckley, 1988; Lee, 1995).

Each model and associated ranking algorithm(s) have their own particular strengths and weaknesses. No one ranking algorithm *always* performs better than an arbitrary, but different, ranking algorithm in *every* situation. This is due in large part to the myriad of applications that a retrieval model and ranking algorithm may have to deal with over a wide spectrum of query-document search models and scenarios. One ranking algorithm, or method, may work well when the document collection is of moderate size and, say, it contains a high percentage of relevant documents for the query submitted to its associated IR (or IRDB) system. Another ranking algorithm may perform well when the document collection is large but not so well when it is small or of moderate size.

Chaudhuri *et al.* provide additional justification for why a single ranking algorithm is inappropriate for all situations, noting that:

> ***1) Flexible scoring and ranking:*** At the heart of a truly versatile DB&IR system is customizable scoring and ranking. Given the wide spectrum of target applications, it is unlikely that a universal best-compromise solution exists. For example, while Page-Rank-style authority measures are a great asset for Web search, they may be meaningless in an intranet setting where authorship and cross-references are tightly controlled; and a journalist working with a news archive every day may want the system to automatically learn scoring weights according to her personal preferences and relevance feedback. At the API level, explicit control over scoring and score aggregation is essential, despite the widespread belief that only ordinal ranks matter; sophisticated applications such as metasearch engines need to distinguish rankings with all scores close to each other from rankings that have wide gaps

> in terms of scores. Also, some applications may wish to produce variable-length result lists by thresholding on absolute scores rather than presenting the top k with a fixed k, if some of the top k results are only marginally relevant. (Chaudhuri et al., 2005)

Suppose that, based on certain parameters and their values, it is possible to determine which of several ranking algorithms will perform better in some situations than in others. Also, suppose that for a particular document collection and query, an IR (or IRDB) system can estimate the values of these parameters. These assumptions, if valid, give the retrieval system the ability to choose the best algorithm in its repertoire for the situation at hand. This was a major goal of this research and represents one of the ways in which the research in this dissertation can be applied.

1.3 Wider Applicability of the Extended Measures

The performance measure equation extensions that were developed in this dissertation for the ASL, MZE, ESL and MRR measures had a wider range of applicability than the settings that they were used in for the dissertation. These measures, and the methodology that was used to develop their associated equations, could also be applied in many settings where the query-document model was different than the one that was used in this dissertation.

The reason for this wider applicability is that the calculations for these extended measures were not directly dependent on a query-document model. Basically, the algorithms that calculated the values for these measures only needed access to two pieces of information for each of the ranked documents: (1) whether or not the document was relevant and (2) the retrieval status value (RSV) for the document. From this information, the algorithms could determine the following information that was needed by the combinatoric models for these extended measures: the number of document equivalence classes, the relative sequence of these classes, the number of documents that each class contained, and the number of relevant documents that each class contained. In addition to this common information, all the performance measure combinatoric models required the document cut-off value and, if the measure was the ESL, also required the requested number of relevant documents. This relevance and RSV information could be obtained efficiently; to collect this information, only a single had to be made over the documents in a vector V of ranked documents.

IR performance evaluation software, like the trec_eval programs (Voorhees and Harman, 2005), often lets the user of that software conflate graded relevance values, or continuous relevance values, to binary relevance values (Kekäläinen and Järvelin, 2002). This was accomplished by establishing a threshold value for the relevance value. Any document that had an RSV that equals or exceeds this threshold value was considered to be a relevant document by the software; otherwise, the document was considered to be a non-relevant document.

1.4 Summary

The general research goal of this dissertation was the use of analytic, as contrasted with retrospective, techniques to construct combinatoric models of IR ranking methods and performance measures for weakly-ordered document collections. These models could be used by researchers to predict system performance, to acquire a deeper understanding of some of the factors that influence how IR performance measures work, and to develop more accurate formulas for these measures. The main items of interest in this research were the Average Search Length, the normalized average position of a relevant document ($\mathcal{A}$), the quality of a ranking method ($\mathcal{A}$), and the development of performance measures that could be calculated at arbitrary points in a vector of ranked documents and that yielded correct results even when the documents were weakly-ordered.

Chapter 2

Background

Retrieval performance measures attempt to provide some indication of how well an information retrieval system performed (if used in a retrospective manner) or is expected to perform (if used in a predictive manner). The Average Search Length is the major measure that is used in this research. Much terminology and concepts appear as part of this research. Definitions of many of them are a part of this chapter. It is important to note that the research that is discussed in this document uses a single term model.

One may naturally wonder "Why is this research limited to just single term queries?" The main reason is that this single term limit "allows us to fully understand many retrieval characteristics and options that are far more difficult to understand in a multi-term case" (Losee, 1998). Another very important reason is that multiple term queries may introduce confounding factors (Johnson and Christensen, 2004) in a research model. If the researcher is not cognizant of these factors, or the factors are not identified and taken into account, then the study may have poor internal validity. A third reason is that many queries, especially on the Internet, consist of just a single term (Jansen et al., 1998). A number of issues may arise with multiple term queries — but can be ignored in the single term case. These include the following issues: If the query terms are not assumed to be independent, then how are term dependencies handled or modeled? Is each query term equally important? If not, how are relative weights specified? Must all of the query terms be present in a document for a match to occur? Do multiple occurrences of a query term mean that they have more weight than a lesser number of occurrences?

Each of the above examples represents issues that have the potential to complicate a retrieval model. The effect of this is that it may hinder the understanding of the characteristics of the information retrieval (IR) model under investigation.

The discussion of the definitions for the terminology and concepts that are used in this research starts by stating that the formula for the Average Search Length (Losee, 1998) is

$$\mathrm{ASL} = N\left(\mathcal{Q}\mathcal{A} + \overline{\mathcal{Q}}\,\overline{\mathcal{A}}\right) + 1/2, \tag{2.0.1}$$

then proceeds by specifying the roles of the independent variables, followed later with a more in depth treatment of these entities. Briefly, N is the number of documents to be ranked, $\mathcal{Q}$ is the probability that the ranking is optimal, and $\mathcal{A}$ is the normalized expected position of a relevant document from the front (i.e., document position 1) of the ranking. In the above formula, $\overline{\mathcal{A}}$ is defined as $1 - \mathcal{A}$ and $\overline{\mathcal{Q}}$ is defined as $1 - \mathcal{Q}$. The values of $\mathcal{Q}$ and $\mathcal{A}$ are in the closed interval $[0, 1]$.

The major part of the process of estimating the ASL involves computing the weighted mean of $\mathcal{A}$ and $\overline{\mathcal{A}}$ with the weights being $\mathcal{Q}$ (the proportion of rankings that are optimal) and $\overline{\mathcal{Q}}$ (the proportion of rankings that are worst-case), respectively.

Hence, given an arbitrary system, its collection of documents, the query, the ranking algorithm, and the collective characterization in terms of N, $\mathcal{Q}$, and $\mathcal{A}$, the expected performance of that system can be calculated. There may be other ways, now and in the future, to estimate the performance of different ranking schemes. They, most likely, will not be exactly identical to the methods which were the subject of this research. However, if someone is interested in doing this kind of performance prediction research, the methods they use will likely have much in common with those used in this research.

Documents with a binary query feature with frequency d may be presented to the user in 1 of 2 distinct orders: *all* the documents with feature frequency d precede *any* document with feature frequency $\overline{d} = 1 - d$ (*optimal ranking*) or vice versa (*worst-case ranking*). Furthermore, it is assumed that the term weight for d is greater than that for $\overline{d}$. In essence, this holds when the query terms are positive discriminators. If the terms are not positive discriminators, then the features must be switched (re-parameterized) so that the product of d and the term weight is greater than the product of $\overline{d}$ and the term weight. If we let $d = 1$, then, in a best-case (or optimal) ranking, all the documents with feature frequency 1 are retrieved before those with feature frequency 0. Likewise, in a worst-case ranking, all the documents with feature frequency 0 are retrieved before those with feature frequency 1.

The mean position, $\mathcal{A}$, on a unit scale, of a relevant document can be computed as the sum of the weighted positions of those relevant documents with feature frequencies d and $\overline{d}$, respectively. These weighted positions are normed to be in the closed interval $[0, 1]$. A document at the front of the ordering has a position of 0 because it is at the low end of the spectrum (good performance), and a document at the back has a position of 1 because it is at the high end (bad performance). $\mathcal{A}$ can be viewed as the expected proportion of all documents that must be examined in the search process to reach the average position of a relevant document in the ordering. It can also be viewed as the mean normalized position of a relevant document in the ordering.

The variable $\mathcal{A}$ is computed by noting that documents with feature frequency d are at the low end of the $\mathcal{A}$ spectrum (good performance) and those with feature frequency $\overline{d}$ are at the high end of the spectrum (poor performance). The formula for $\mathcal{A}$ is

$$\mathcal{A} = \frac{1 + \Pr(d) - \Pr(d|rel)}{2}. \tag{2.0.2}$$

Notationally, the equation can be simplified by letting $p = \Pr(d|rel)$ and $t = \Pr(d)$:

$$\mathcal{A} = \frac{1 + t - p}{2}. \tag{2.0.3}$$

A ranking is an ordering or sequencing. With respect to the ranking of documents, in response to a query, an optimal ranking is a sequence where the documents that contain the query term are at the front of the sequence and any that do not contain the term appear after the last document that contains the term in that sequence. A worst-case ranking is the polar opposite (i.e., all of the documents that contain the term are at the rear of the sequence, all of the other documents are at the front). A random-case ranking is a sequencing where it is equally likely for any document, whether or not it contains the term, to occupy an arbitrary position in that sequence.

2.1 Several Alternative Measures That Are of Interest

Of course, the ASL measure is far from the only measure that can be used to help assess ranking performance. Some of the many other measures are the Expected Search Length (ESL) (Equation 2.1.1 on the following page), the Mean Reciprocal Rank (MRR) (Equation 2.1.3 on page 17), and the MZ-based E measure (MZE) (Equation 2.1.4 on page 17). These three measures are of great interest for the last of the three research questions being addressed by this dissertation. The discussion for this third research question takes place in Chapter 10 (The ASL Measure and Three Frequently-Used Performance Measures).

In Chapter 10, combinatoric-based models are developed for each of these three measures, and for the Average Search Length (ASL) measure. These models provide an analytic way to calculate the values of these measures and are very prominent in the discussions that occur in Chapter 10.

2.1.1 Expected Search Length

The ESL (Cooper, 1968) is similar to the Average Search Length. The major difference is that it counts the mean number of non-relevant documents retrieved *before* the kth relevant document is retrieved in a rank-ordered vector V of documents. In other words, it counts the mean number of non-relevant documents retrieved in order to produce a given number k of relevant documents. For a query q, a vector V of ranked documents, and a request for the first x relevant documents, the ESL can be defined as

$$\mathrm{ESL}(V, x) = j + \frac{i \cdot s}{r + 1}, \tag{2.1.1}$$

where l is the level at which the xth relevant document occurs, j is the total number of documents irrelevant to q in all levels which precede level l in the weak ordering, i is the number of documents irrelevant to q in level l, s is the number such that the sth relevant document found in level l of the weak ordering would complete the search for request q, and r is the number of documents level l which are relevant to q.

Caution must be taken when referring to the Expected Search Length (ESL), though, because Cooper's definition is not universally used (Korfhage, 1997). Some researchers in the IR community have defined the ESL to be the mean number of total documents (i.e., both relevant and non-relevant) retrieved in order to obtain the xth relevant document in a rank-ordered vector V of documents. In other words, this alternative ESL definition counts the mean number of total documents retrieved in order to produce a given number x of relevant documents. For example, if the user requests 6 relevant documents and a mean of 4 non-relevant documents are retrieved before the sixth relevant document is retrieved, the Cooper version of the ESL calculates the mean number of retrieved documents as 4 documents whereas the alternate version considers the mean to be 10 documents.

2.1.2 Mean Reciprocal Rank

There are several performance evaluation measures in IR that are based on the concept of *reciprocal rank* (RR). The most well-known one is the *mean reciprocal rank* (MRR). It is used very heavily in the TREC Question Answering (QA) tracks (Voorhees and Tice, 1999; Voorhees, 1999) to assess the performance of an IR system on a set of questions Q.

More formally, the reciprocal rank at document cut-off value k on a rank-ordered vector V of answers is defined as

$$\mathrm{RR@}k(V) = \begin{cases} 1/i, & \text{if } \exists i \leq k, \text{ such that } V[i] \text{ is a correct answer, and} \\ & \quad \forall j < i, V[j] \text{ is an incorrect answer;} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1.2}$$

The above expression indicates that if a correct answer occurs among the first k answers in a rank-ordered vector V of answers, then the expression's value is the reciprocal of the rank that corresponds to the first correct answer. If there is no correct answer among the first k answers, then the reciprocal rank is defined to be 0. For example, assume that $k = 5$ and that correct answers are at ranks 2 and 3. Then the reciprocal rank is 1/2 because the first correct answer was at rank 2. Now, assume that the first correct answer is at rank 7. In this case, the reciprocal rank is 0 because the first correct answer was at a rank that is greater than 5.

According to Lin et al. (2008), two commonly used measures of a QA system's performance are "the top-1 accuracy and the top-5 mean reciprocal rank." The top-1 accuracy for a question set Q is the proportion of correct answers that are at rank 1 for the questions in Q. It is defined as

$$\text{top-1 accuracy} = |\{q | q \ \in Q \text{ and } \ V_q[1] \text{ is a correct answer}\}| / |Q|,$$

where V_q is a rank-ordered vector of answers for question q. The mean reciprocal rank at document cut-off k for a vector V of answers is defined as

$$\mathrm{MRR}(Q)@k(V) = \frac{\sum_{q \in Q} RR@k(V_q)}{|Q|}, \tag{2.1.3}$$

where Q is a set of questions, $q \in Q$, and V_q is the rank-ordered vector of answers for question q. Expressed another way, the MRR is the mean of the reciprocals of the ranks of the first correct answer that occurs among the top k (in TREC, $k = 5$) answers in a ranking for each question. Note that the sets of answers represented by V and V_q are identical.

2.1.3 MZ-Based E Measure

This measure is based on measurement theory (Bollmann and Cherniavsky, 1981) (as contrasted to Swets' E measure which is based on the Receiver Operating Characteristics (ROC) model (Swets, 1969; Pepe, 2003)).

This measurement theory version of the E measure (MZE) (van Rijsbergen, 1979; Baeza-Yates and Ribeiro-Neto, 1999; Manning et al., 2008) is defined as

$$\mathrm{MZE} = 1 - \frac{2}{P^{-1} + R^{-1}}, \tag{2.1.4}$$

where P represents precision and R represents recall.

2.2 Mathematical Presentation and Techniques

This research made use of mathematical proofs, probability theory, probability models, simulation, and combinatorial enumeration algorithms. Below are brief descriptions of each accompanied by remarks with respect to their various advantages, disadvantages, and limitations.

2.2.1 Notation

This dissertation used mathematical notation, some of which may be unfamiliar to its readers. The List of Symbols that starts on page xxxiii contains the symbols and constructs that are widely used in this dissertation. The logarithm is the natural logarithm (i.e., $\log(x) \equiv \log_e(x)$, where $e = 2.71828...$). In practice, the logarithm base could just as easily have been 2, 10, 16, or some other positive number greater than 1, because a logarithm in one base can always be transformed to one in another base by multiplying it by a constant that is specific to the two bases.

2.2.2 Proofs

"[P]roofs play a central role in mathematics [and in mathematics-related portions of many of the sciences], and deductive reasoning is the foundation on which proofs are based" (Velleman, 1994). The proofs that appeared in this dissertation came almost exclusively from Chapter 5 (calculating $\mathcal{Q}$ for the coordination level matching (CLM) ranking method), Chapter 6 (calculating $\mathcal{Q}$ for the inverse document frequency (IDF) and decision-theoretic (DT) ranking methods), Chapter 7 (calculating $\mathcal{A}$ and the ASL), and Chapter 8 (formula validation). Many of the concepts that were introduced in these 3 chapters needed to be rigorously established. Lemmas (i.e., conjectures) were associated with these concepts and the validity of each lemma was established by a proof.

When performing research, one often observes patterns and relationships among the various entities that are being studied. These observations lead to conjectures about the relationships. The only way one can be sure that such a relationship is true, is by presenting a valid mathematical proof. Essentially, "a mathematical proof is a convincing argument that starts from the premises [statements assumed to be true], and logically deduces the desired conclusion" (Bloch, 2000). Two strengths of a proof are that (1) unlike a theory or hypothesis, it is not falsifiable and that (2) the derivation of a proof can provide additional insight into a problem. One of the limitations of a proof is that it is only available within the realm of precisely defined mathematical constructs. Its power beyond those strictures depends upon the closeness with which the mathematics models the entity being analyzed.

2.2.3 Probability Theory and Models

Probability theory is the branch of mathematics that deals with the analysis of random events. One of its main uses, in the research contained in this document, was to construct probability models. A probability model is a *scientific model* that incorporates uncertainty. These types of models are also known as *stochastic models*.

> A scientific model is an abstract and simplified description of a given phenomenon. Certain basic aspects of this phenomenon are isolated as being of primary interest, and an analogy is drawn between these aspects and some logical structure concerning which we already have detailed information. Scientific models are most often based on mathematical structures ...
>
> When an investigator builds a mathematical model for a particular natural phenomenon (say, the motion of an asteroid), important elements of this phenomenon (the position, mass, shape, and speed of the asteroid) are identified with the basic elements of some mathematical structure (numbers). Certain fundamental facts connecting the important elements of the phenomenon are restated as axioms relating the analogous mathematical entities. Finally, the more complex relationships between the basic elements of the natural phenomenon are made to correspond to laws or theorems in the mathematical structure. If this correspondence is reasonably valid, the investigator does not have to experiment haphazardly with the phenomenon to find new facts; instead, logical arguments based on the mathematical axioms can lead to a theorem that presumably has an analogy to a law of nature. Experimentation can now be directed toward verifying this law.
>
> The fact that an investigator need only concentrate on the few axioms that define the mathematical structure of his model leads to a simplification and unification of his knowledge concerning the natural phenomenon. Every fact known to him can be reproduced by starting from the axioms and using mathematical logic. Thus, his discipline becomes a cohesive whole in which all facts are logically interrelated, rather than merely a list of isolated facts. (Olkin et al., 1994)

The above quote describes the concepts of scientific and mathematical models. Basically, a model is an abstraction of some real world phenomena where the relationships between the various parts can be modeled mathematically. Often, mathematical rules, or axioms, can be developed to manipulate and study parts of the model. Many characteristics of probabilistic models can be determined analytically. However, when these models are too complex or intractable for analytical treatment, simulation methods are often used to help answer questions about the phenomena being studied.

2.2.4 Simulation

Simulation is used in Chapter 8 to help estimate the quality of ranking value for large document collections in situations where it is infeasible to use brute force techniques to determine these values.

> The Latin verb *simulare* means to imitate or mimic. The purpose of a simulation model is to mimic the real system so that its behavior can be studied. The model is a laboratory replica of the real system, a microworld (Morecroft, 1988). By creating a representation of the system in the laboratory, a modeler can perform experiments that are impossible, unethical, or prohibitively expensive in the real world. (Sterman, 1991)

The quote above describes a simulation model and how such a model can be used to study real systems that may be impractical, or impossible, to study or manipulate by other means. More specifically, simulation can also be used to study information retrieval problems and is an alternative to the experimental approach so prevalent in IR research. Heine (1981) labels this type of approach a “simulation experiment.” Paraphrasing Cooper (1971a), these are the 4 situations in which simulation can be a useful tool: the situation in which it is desired to modify a system that cannot, in practice, be modified; the situation in which it may be possible to modify the system and observe the result, but the cost to do so may be prohibitive; the situation in which the system is so complex it cannot be described in an analytical form; and the situation in which a system can be described analytically but cannot be solved analytically.

According to Law (2006), the main advantages of simulation are that it allows arbitrary model complexity; it circumvents analytically intractable models; it facilitates what-if and sensitivity analyses; the process of building a model can lead to system improvements and greater understanding; and it can be used to verify analytic solutions. The main disadvantages of simulation are the following: it provides only solution estimates; it only solves one set of parameters at a time; and it can take a large amount of development and computer time.

2.2.5 The Query-Document Model

In the query-document model that was used in this dissertation, a query consisted of a single term and each document contained at least one term. The query term may, or may not, be contained in a document. Multiple occurrences of a term in a document have no more significance than a single occurrence of the term. A document is either relevant or non-relevant to a query; that is, the model uses binary relevance.

A particular query and an associated document collection of cardinality N was modeled in this research by a set of ordered arrangements of nonnegative integers. Each ordered arrangement was a sequence of $k > 0$ natural numbers that summed to N. These were known as *weak compositions* of size k (i.e., *weak k-compositions*). In this dissertation, the value of k was almost always 4. However, there were a few instances where k had the value 2 (Sections 5.8.5 and 8.3.1) or where k had the value 3 (Section 8.3.1).

Weak compositions are important to this research because they were used to aid in the construction of, and reasoning about, some of the performance models that are studied. The next section contains a detailed discussion of weak compositions and relates them to this query-document model.

2.2.6 The Relationship Between the Query-Document Model and Weak 4-Compositions

We start this section by providing detailed information about weak and strong compositions. After that, we discuss how weak and strong 4-compositions can be used to represent the query-document model that was used in this dissertation.

If each of the k numbers in an ordered arrangement (such as the type of arrangement that is introduced in Section 2.2.5) must be positive, then the arrangement is not only a weak k-composition, but is also a *(strong) k-composition.* The set of (strong) k-compositions is a proper subset of the set of weak k-compositions. Figure 2.1 on the following page depicts the relationship between sets of weak compositions and sets of compositions. From this point on, (strong) compositions are generally referred to as simply compositions unless the author wants to contrast a (strong) composition with a weak one. The notation $[k]$, used in the quote below from Bóna (2006), denotes the set of the first k positive integers, that is, [5] represents the set $\{1, 2, 3, 4, 5\}$.

More formally, here are definitions for weak compositions and compositions:

> A sequence (a_1, a_2, ..., a_k) of integers fulfilling $a_i \geq 0$ for all i, and (a_1 + a_2 + ... + a_k) = n is called a *weak composition* of n. If, in addition, the a_i are *positive* for all $i \in [k]$, then the sequence (a_1, a_2, ... , a_k) is called a *composition* of n. (Bóna, 2006)

For example, the compositions of size 4 of the number 5 are (1, 1, 1, 2), (1, 1, 2, 1), (1, 2, 1, 1), and (2, 1, 1, 1). An alternative way of viewing them is as ordered sums:

$$\begin{aligned} 5 &= 1 + 1 + 1 + 2 \\ &= 1 + 1 + 2 + 1 \\ &= 1 + 2 + 1 + 1 \\ &= 2 + 1 + 1 + 1. \end{aligned}$$

The weak compositions of size 2 of the number 3 are (0, 3), (1, 2), (2, 1), and (3, 0).

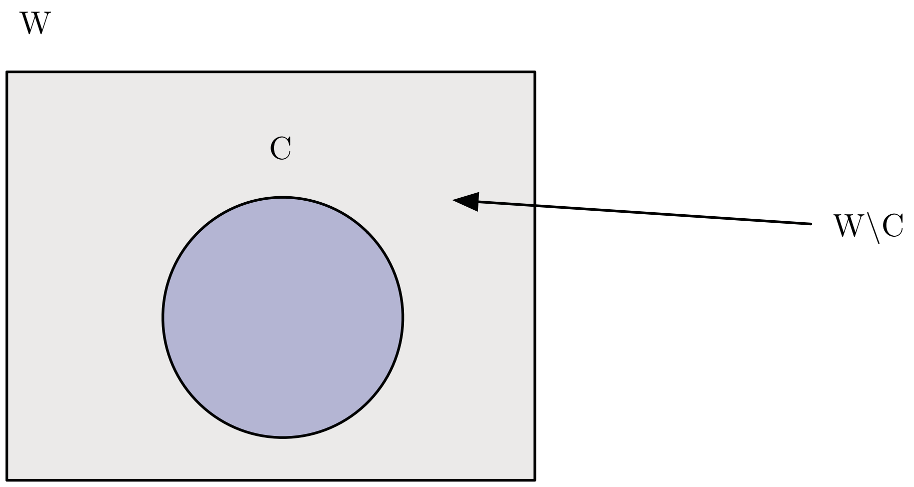


Figure 2.1: The relationships between the sets of compositions (C) and weak compositions (W) for a positive integer n into k parts. The circle represents the set of compositions and the backslash ($\backslash$) symbol denotes set complementation. The set $W \backslash C$ denotes the weak compositions that are not simultaneously compositions. That is, the set $W \backslash C$ denotes the weak compositions that are not members of set C. The gray region represents the members of $W \backslash C$.

An alternative viewing is:

$$\begin{aligned} 3 &= 0 + 3 \\ &= 1 + 2 \\ &= 2 + 1 \\ &= 3 + 0. \end{aligned}$$

Now, let us imagine that we have a collection of N documents and a particular single-term query. Furthermore, let us assume that, for each document, we are interested in two pieces of information: whether that document is relevant to the query and whether its bag of terms contains the query term. This divides the document collection into 4 non-overlapping (i.e., mutually exclusive) categories: the documents that are relevant and contain the query term (r_1 denotes the cardinality of this category), the documents that are relevant but do not contain the query term (r_0 denotes the cardinality of this category), the documents that are non-relevant and contain the query term (s_1 denotes the cardinality of this category), and the documents that are non-relevant and do not contain the query term (s_0 denotes the cardinality of this category).

Each of these categories contains anywhere from none to all of the documents in the collection. No matter how many documents each category contains, though,

$$r_0 + r_1 + s_0 + s_1$$

must always equal N because each document must be a member of exactly one of these 4 categories. Notationally, let

$$N = R + S = n_0 + n_1$$

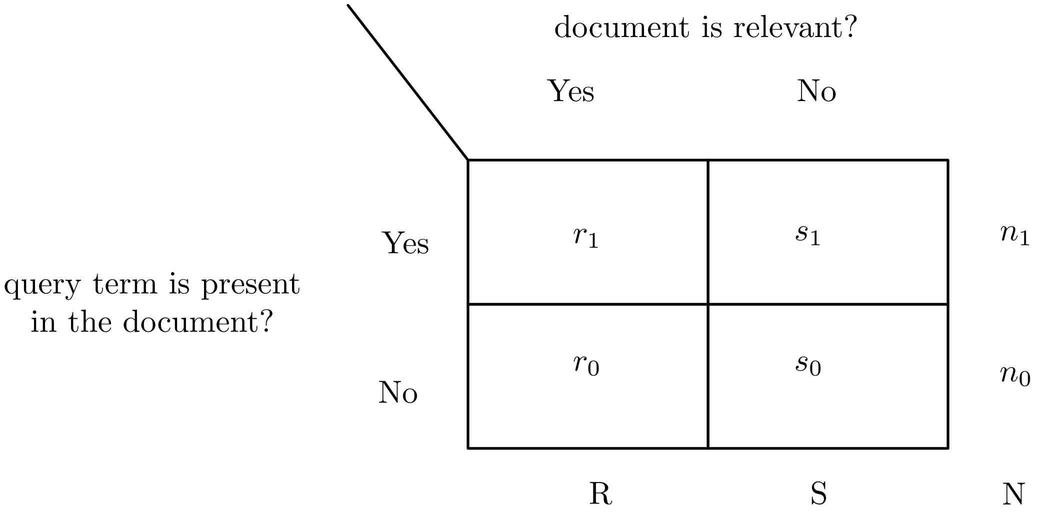
represent the total number of documents in a collection with

$$R = r_0 + r_1$$

representing the number of *relevant* documents and $S = s_0 + s_1$ representing the number of *non-relevant* documents. Figure 2.2 on the next page uses a contingency table to depict the relationships between these variables.

The above requirements are very naturally modeled by a set of weak compositions of size 4 of N. Each weak composition is represented by the following ordered arrangement: (r_1, s_0, r_0, s_1). There is nothing special about this particular arrangement, the sequence above is just one of $4! = 24$ different ways that we could have arranged those 4 distinct symbols. Two of the remaining 23 possibilities are (r_0, r_1, s_0, s_1) and (r_0, s_0, r_1, s_1).

Essentially, each weak composition corresponds to one way that a collection of N



		document is relevant?		
		Yes	No	
query term is present in the document?	Yes	r_1	s_1	n_1
	No	r_0	s_0	n_0
		R	S	N

Figure 2.2: The relationships discussed earlier between N, R, S, r_0, r_1, s_0, s_1, n_0, and n_1 can be succinctly expressed by this 2x2 contingency table.

documents can be divided into 4 non-overlapping (i.e., mutually exclusive) categories. The set of weak compositions for a particular query and an associated document collection of cardinality N represents all of the unique ways that N documents could be assigned to the 4 categories just mentioned above. How to calculate the cardinality of this set is discussed below.

A primary item of interest in some of the modeling scenarios that this research explored was the sample space of weak compositions for an N-document collection. More specifically, the interest was in the generation of the sample space and the number of weak compositions in this space whose elements satisfied particular mathematical constraints. This research mainly used weak compositions of size 4 to help determine probabilities or proportions in various modeling scenarios.

In IR terms, a weak composition of size 4 is a collection of N documents where *at least one* of the following conditions must be true: the number of relevant documents that contain the query term is 0 (i.e., $r_1 = 0$), the number of relevant documents that do not contain the query term is 0 (i.e., $r_0 = 0$), the number of non-relevant documents that contain the query term is 0 (i.e., $s_1 = 0$), or the number of non-relevant documents that do not contain the query term is 0 (i.e., $s_0 = 0$).

Also, in IR terms, a composition of size 4 is a collection of N documents where *all* of the following conditions must be true: the number of relevant documents that contain the query term is positive (i.e., $r_1 \geq 1$), the number of relevant documents that do not contain the query term is positive (i.e., $r_0 \geq 1$), the number of non-relevant documents that contain the query term is positive (i.e., $s_1 \geq 1$), and the number of non-relevant documents that do not contain the query term is positive (i.e., $s_0 \geq 1$).

According to Bóna (2006) and Weisstein (2003), the number of compositions of n into k parts is given by

$$C_k(n) = \binom{n-1}{k-1} \tag{2.2.1}$$

and the number of weak compositions of n into k parts is given by

$$\tilde{C}_k(n) = \binom{n+k-1}{k-1}, \tag{2.2.2}$$

where $\binom{n}{k}$ denotes the number of combinations of n things taken k at a time, $C_k(n)$ denotes the number of compositions of n into k parts, and $\tilde{C}_k(n)$ denotes the number of weak compositions of n into k parts. Figure 2.1 on page 23 illustrates an important relationship between the set of weak compositions of n into k parts and the set of compositions of n into k parts.

Related symbols that are used later in this work are $C(n, k)$ (an alternate notation for $\binom{n}{k}$), $P(n, k)$ to denote the number of permutations of n things taken k at a time, and $n!$ to denote the number of permutations of n distinct objects.

The first identity above (i.e., Equation 2.2.1) provides a way to determine the cardinality of the sample space when each integer in a composition must be at least 1. The second identity (i.e., Equation 2.2.2) calculates the cardinality when an integer is allowed to be 0. The latter identity is expected to be of more use in this research mainly because any of the 4 integers in an ordered arrangement of 4 integers for a modeling scenario could have the value 0. For example, the weak composition $(r_1, s_0, r_0, s_1) = (1, 5, 0, 3)$ represents a nine (e.g., $1 + 5 + 0 + 3 = 9$) document collection that has 1 relevant document where the query term is present, 5 non-relevant documents where the term is absent, 0 relevant documents where the term is absent, and 3 non-relevant documents that have the term present.

2.2.7 Combinatorial Generation and Enumeration Algorithms

Basically, enumeration is simply counting. In this research, we were primarily interested in the use of generation and enumeration algorithms to help validate some of the combinatoric formulas that were derived as part of the combinatoric-based versions of the performance models for the ASL. Most of the validation-related discussions in this dissertation occur in Chapter 8 (Validation of the Formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ Measures). Section 10.8 discusses the validation of the versions of the performance measures that were derived in Chapter 10 and calculate correct results even when a collection of documents for a query is weakly-ordered. In Section 10.8, we introduce the notions of *Type-T* and *Type-D* performance measures. Briefly, we use *Type-T* as an adjective to denote a performance measure whose calculated values are consistent with the assumption that *some* of the documents in a vector V of ranked documents *may* have tied (i.e., duplicate) RSVs. And, we use *Type-D* as an adjective to denote a performance measure whose calculated values are consistent with the assumption that *all* the documents in a vector V *must* have distinct RSVs.

From the discussion about the query-document model in Section 2.2.5, we can view a weak 4-composition (r_1, r_0, s_1, s_0), for an N document collection C, where

$$N = r_1 + r_0 + s_1 + s_0,$$

as one of the $\binom{N+3}{3}$ unique ways that a query q could partition this collection into 4 mutually distinct parts. Any query q always maps to exactly one of these weak 4-compositions. No matter how much time, energy, and ingenuity a user has, that user cannot construct any more than $\binom{N+3}{3}$ unique queries (from the viewpoint of our query-document pair model). As a simplification, any weak 4-composition (r_1, r_0, s_1, s_0) can be thought of as *a* "query" for a document collection of size $N = r_1 + r_0 + s_1 + s_0$.

The discussion below is applicable to the combinatorial model that is detailed in Section 3.3. In this model, we are interested in counting the number of weak compositions in the sample space that satisfy certain constraints. The question is "How do we do this?" In some cases, it may be easy to do analytically. In others, we may still be able to do it analytically, but with the expenditure of a lot more effort and possibly some ingenuity. An alternative way is to generate all the weak compositions in the sample space, then count the ones that satisfy the constraints. Another technique would be to count the qualifying weak compositions, but not to generate them. To help put this in context, and to provide more background, combinatorial algorithms are discussed below.

According to Kreher and Stinson (1999), combinatorial algorithms exist to investigate combinatorial structures. They are informally classified according to their main purpose: *generation* – construct all the combinatorial structures of a particular type; *enumeration* – compute the number of different structures of a different type; and *search* – find at least one example of a particular type (if it exists).

Every combinatorial generation algorithm can be trivially modified to also be a combinatorial enumeration algorithm; however, the converse is not true (Kreher and Stinson, 1999). The modification enhancement is to just add statements to count each combinatorial structure as it is being generated and to output this tally at the completion of the generation process.

Analytic formulas can often be used as an alternative to counting via combinatorial algorithms. The branch of combinatorics associated with the derivation and application of these formulas is known as *enumerative combinatorics* (Benjamin and Quinn, 2003; Bóna, 2007; Charalambides, 2002, 2005; Goulden and Jackson, 1983). However, in many situations, these analytic formulas can be difficult to derive or are intractable with respect to manipulation. This is the area where combinatorial generation and enumeration algorithms are often of great help.

If counting is the sole reason for using these algorithms, an enumeration algorithm is preferred over a generation one, because "it is often easier to enumerate the number of combinatorial structures of a particular type than to actually list them" (Kreher and Stinson, 1999). A particular weakness of all combinatorial algorithms – in particular, combinatorial generation algorithms – is that most combinatorial problems are big. Often, due to practical constraints such as computer memory size, processor speed, disk storage requirements, and the computational complexity of the algorithm, the size of the problem being investigated by a combinatorial algorithm has to be restricted.

2.3 Term and Query Operations

This section describes several of the basic operations that were used to decompose the queries and documents, that were associated with the Cystic Fibrosis (CF) test collection, into tokens. Stoplists were applied to these tokens and the remaining terms were stemmed. The resultant terms were later used to help construct a modified version of the CF test collection where the original multiple term queries were transformed into single term queries by a process that is described in Section 2.6.

These operations were necessary because the queries and documents that were associated with a collection (e.g., the CF test collection) were generally not very useful in their raw forms. These queries and documents typically needed to undergo several stages of preprocessing in order to change them from their raw form into a form that was much more amenable for the kinds of performance studies that occurred in this dissertation. Basically, preprocessing decomposed the terms in the queries and documents into terms (words) that were later normalized (via stemming) after non-content terms (e.g., a, the, of, an) were eliminated by the use of a stoplist.. The remainder of this section provides more detail about these operations.

2.3.1 Lexical Analysis

Lexical analysis is the first stage of automatic indexing and of query processing. It is used to analyze a document (or query) to determine its terms and to decompose the document (or query) into these terms. A software construct known as a *lexical analyzer* implements this stage. Basically, a lexical analyzer breaks text into terms. A term may be a word or a sequence of words (to be discussed subsequently).

> There are three ways to implement a lexical analyzer:
>
> - Use a *lexical analyzer generator*, like the UNIX tool `lex` (Lesk, 1975), to generate a lexical analyzer automatically;
> - Write a lexical analyzer by hand *ad hoc* [emphasis added]; or
> - Write a lexical analyzer by hand as a finite state machine. (Frakes and Baeza-Yates, 1992)

Which way is best depends on the situation. If the lexical analyzer is complicated, then the first way is the best; if the lexical analyzer is simple, then handcrafting the lexical analyzer (i.e., the second and third ways) may be the implementation technique of choice. The latter of the handcrafted ways is superior to the other one because the *ad hoc* implementation of the lexical analyzer is much more likely to contain errors and inefficiencies. In this research, a lexical analyzer generator was used to produce the lexical analyzer.

Conceptually, a term in this research corresponded to a single word. This begs the question: *What counts as a word or token in either the query or a document?* An easy reply is that terms consisting solely of letters should be words or tokens. However, Frakes and Baeza-Yates (1992) indicates that “problems soon arise, however.” Some questions related to potential problem areas are: Is a string of digits a token or should a token possibly contain digits? Should hyphenated words be broken into their constituents? Should other punctuation (e.g., commas, periods) be part of a token? Is the case of letters of any significance in a word?

In greater detail, Frakes and Baeza-Yates (1992) lists these as potential issues:

- Digits—Most numbers do not make good index terms, so often digits are not included as tokens. However, certain numbers in some kinds of databases may be important (for example, case numbers in a legal database). Also, digits are often included in words that should be index terms, especially in databases containing technical documents. For example, a database about vitamins would contain important tokens like “B6” and “B12.” One partial (and easy) solution to the last problem is to allow tokens to include digits, but not to begin with a digit.
- Hyphens—Another difficult decision is whether to break hyphenated words into their constituents, or to keep them as a single token. Breaking hyphenated terms apart helps with inconsistent usage (e.g., “state-of-the-art” and “state of the art” are treated identically), but loses the specificity of a hyphenated phrase. Also, dashes are often used in place of ems, and to mark a single word broken into syllables at the end of a line. Treating dashes used in these ways as hyphens does not work. On the other hand, hyphens are often part of a name, such as “Jean-Claude,” “F-16,” or “MS-DOS.”
- Other Punctuation—Like the dash, other punctuation marks are often used as parts of terms. For example, periods are commonly used as parts of file names in computer systems (e.g., “COMMAND. COM” in DOS), or as parts of section numbers; slashes may appear as part of a name (e.g., ”OS/2”). If numbers are regarded as legitimate index terms, then numbers containing commas and decimal points may need to be recognized. The underscore character is often used in terms in programming languages (e.g., “max_size” is an identifier in Ada, C, Prolog, and other languages).
- Case—The case of letters is usually not significant in index terms, and typically lexical analyzers for information retrieval systems convert all characters

> to either upper or lower case. Again, however, case may be important in some situations. For example, case distinctions are important in some programming languages, so an information retrieval system for source code may need to preserve case distinctions in generating index terms. (Frakes and Baeza-Yates, 1992)

The lexical analysis in this research adhered to the following choices: case was insignificant; any nonempty string of letters and digits, not beginning with a digit, was regarded as a term; and all punctuation, spacing, and control characters were treated as term delimiters.

2.3.2 Stoplists

Many of the terms that appear in a typical document are not good for indexing. Often, these are terms that can be ignored because their discrimination value is marginal at best. A stoplist contains a list of such terms. Salton and Smith (1989) proves that not all terms are equally good for indexing. They provided a mechanism for selecting good terms.

Sparck-Jones and Galliers (1996) says that the goodness of index terms had to be evaluated in an indirect (i.e., extrinsic) way. Basically, the terms' quality is measured by how well they perform with respect to some *other* task. The performance measures typically used are recall and precision.

Fox (1992) states that lexical analysis starts after the text has been processed and stored. Here, the purpose is to take a stream of text and convert it into tokens. Many of these tokens became candidate index terms. Later, after additional processing, a significant number of these become actual index terms.

A *function word* is a word such as a preposition, article, auxiliary, or pronoun, that chiefly expresses grammatical meanings and has little semantic content of its own (Webster's, 1996). This is in contrast to a *content word*, that carries semantic content, bearing reference to the world independently of its position in a sentence (Webster's, 1996).

Salton (1975) recognizes that certain high frequency words were not content-bearing and, thus, have no positive effect on index term selection. The notion of a stoplist was developed to exclude these words from being index term candidates.

Various techniques have been proposed in the literature for stoplist construction (Loos et al., 2005). One way might be termed the "word class" approach. This approach recognizes that certain classes of words are better content indicators for a document than others. There are two strategies within this approach. One is to build a *generic stoplist* (Hoch, 1994). This list consists of function words. An opposite strategy is to designate words that fall into certain syntactic classes as content-bearing and to only use these as index term candidates (Luhn, 1957; Prikhod'ko and Skorokhod'ko, 1982).

Another technique is to include the most frequently occurring words in a stoplist (Luhn, 1957; Salton and Smith, 1989). According to Moens (2000), there are two variations on this. The first is to construct a generic stoplist by analyzing a general corpus (e.g., the Brown Corpus of Standard American English (Wikipedia, 2006)) for the most frequently used words. The second is to construct a *domain-specific stoplist*. This stoplist just focuses on words in the subject area in the domain that the indexing is intended to take place in. No matter which variation is chosen, a value is chosen to either specify the maximum cardinality of the stoplist or to define the minimum frequency of occurrence for a list entry (Moens, 2000).

Since function words tend to have a small number of characters, a shortcut that is used by some indexing software implementations considers any word at or below a fixed number of characters to be a function word (Ballerini et al., 1996). However, because a content word (e.g., mob) could be removed with this scheme, an *anti-stopword list* (Knaus et al., 1995) can be created to prevent the unwanted exclusion of short content words as index terms.

A more sophisticated technique for identifying domain-specific stopwords uses machine-learning techniques. This technique "uses a collection of *training texts* and information about their *relatedness* in the training set (Wilbur and Sirotkin, 1992; Yang and Wilbur, 1996)."

2.3.3 Stemming

There are several varieties of stemming. One commonly used variation normalizes terms mainly by eliminating variations in their prefixes and suffixes. This form and several others are discussed in the subsequent paragraphs. No matter what variation of stemming is used, the main purpose of any variation is to standardize the representation of term variants. This enhances the chances of matching similar terms (i.e., increases recall) but blurs the distinction between individual terms (i.e., decreases precision).

> Stemming in the field of information retrieval aims at *improving the match between the index terms of query and document text.* The chances of matching increase when the index terms are reduced to their word stems. Stemming, thus, is a recall-enhancing device to broaden an index term in a text search (Salton, 1986). Additionally, stemming reduces the number of index terms by mapping the morphological variants to a standard form. Consequently, the size of the text representation decreases, which is beneficial in terms of storage. (Moens, 2000)

In addition to the above description of stemming , Moens (2000) states that "[t]here are four automatic approaches to stemming." Those approaches are *table lookup* method, *affix removal algorithms*, *letter successor variety stemmers*, and the *n-gram method.*

The table lookup method is the simplest of the 4 approaches. It mandates that a term and its corresponding stem be stored as a pair in a table or dictionary (Frakes and Baeza-Yates, 1992).

Affix removal stemmers work by deleting prefixes, suffixes, or both, from a term in order to reduce it to a stem (Frakes and Baeza-Yates, 1992). Some algorithms may also transform the resultant stem. According to Moens (2000), stemmers use this approach more than any of the others. The affix removal stemmers that are most frequently cited in the literature are the Porter stemmer (Porter, 1997), the Lovins stemmer (Lovins, 1968), the Krovetz stemmer (Krovetz, 1993), and the Lancaster Paice/Husk stemmer (Paice, 1990). The algorithms they employ are heavily dependent on which language they were written to handle. This is not just a characteristic of the those three stemmers, but of any affix removal stemmer. Frakes and Fox (2003) is a recent study, using various measures, that compared how well many of these stemmers perform. The findings from the study state that that these stemmers have various strengths and weaknesses and that their differences are statistically significant.

Letter successor variety stemmers (Hafer and Weiss, 1974) use the frequencies of letter sequences in a body of text as the basis of stemming.

The n-gram method conflates terms based on the number of consecutive characters they share (Frakes and Baeza-Yates, 1992). If $n = 2$, the consecutive characters are called *digrams*; if $n = 3$, they are called *trigrams*. Even though this method is grouped with the "stemming" approaches, in actuality, it is not a stemming technique because no stem is produced. Essentially, this is a statistical procedure that evaluates the n-grams to see which of them are most similar to those that exist in the n-grams derived from the index database for the corpus.

2.4 An Historical Overview of Information Retrieval Research

Information retrieval and the associated research involving it, from the mid-1940s up to the present day, have been heavily influenced by the times that they were, or are, a part of and the technology and resources available during those times. It would be remiss to review the literature of research methodology and methods used in information retrieval research and experimentation without also commenting about how technology and other factors have changed the research focus in IR over time.

Two major factors in the 1950s contributed to the beginning of IR research. The first factor was that, at the end of WWII, many "scientific, technical, and patent documents generated during and shortly after the war necessitated new approaches to organizing, controlling, storing, retrieving, and accessing documents. Further-more [*sic*], traditional classification schemes were not sufficiently discriminating to deal with the rapid growth of the scientific, medical, and legal journal literature" (Griffiths and King, 2002). Even though the number of documents released would not be overwhelming by today's standards, at that time, they greatly taxed the resources available to process them. More importantly, those resources were more attuned to processing numeric data rather than textual information. This shortcoming sparked research into finding effective methods, tools, and techniques for the indexing of, and the search for, documents. The second factor was related to the increasing use of computers for processing repetitive tasks. In was quickly recognized that computers could assist with the representation, storage, retrieval, and classification of documents.

During the first period, which lasted up until around the mid-1970s, IR research was concerned with the improvement of search engines (using today's terminology) for scientific literature. The major emphasis was on the development and improvement of computer algorithms so that they could better and more efficiently handle the great amount of electronic data and information resources (Kagolovsky, 2003). This period "focused mostly on experimentation and evaluation that attempted to address IR systems inherent weaknesses. In particular, searching during these early phases exhibited slow system response times and expensive human intervention" (Griffiths and King, 2002). Since no IR system is perfect, they all contain flaws (i.e., they retrieve documents that are not pertinent (non-relevant) to a query and miss pertinent (relevant) ones). Griffiths and King (2002) says that "[i]n fact, a great deal of effort and controversy in the 1950s and 1960s focused on developing measures and methods for IR experimentation and evaluation. Much of IR research and design from the 1950s through the 1970s aimed at reducing these errors".

Information retrieval systems then were considerably different than the ones that we encounter today. Two particularly telling comments about the strictures that IR systems and researchers had to operate under during that period (but not in the later periods) are those below:

> These first systems were hampered by the limited processing power of early computers, and the limited capacity for and high cost of storage. They operated offline, in a batch processing mode. It was not until the 1970s that IR systems made it possible for users to submit their queries and obtain an immediate response, allowing them to view the results and modify their queries as needed. The development of magnetic disk storage and improvements in telecommunications networks at this time made it possible to provide access to IR systems nationwide. (Rasmussen, 2005)

and

> Computers were expensive, difficult to operate, and not very user-friendly. They were in the hands of engineers, and potential users did not interact with computers directly. Queries were submitted to the intermediate person, searches were performed in batches, and answers could take days. In the l970s, information systems were still not powerful enough to store large databases, and were only able to work with bibliographic databases. As a result, research was focused primarily on development and improvement of techniques for storing and retrieving text documents. (Kagolovsky, 2003).

As a consequence, the research focus was on system issues such as the development and improvement of algorithms and storage and retrieval techniques. The user was not a major concern during this period. However, even during this systems-focused period, researchers were beginning to realize that information retrieval needed to start incorporating the user into its experiments and studies. This led to a more encompassing view of IR and the start of the second period.

The second period, from about 1975-1985, began when researchers increasingly saw the need to make the user an integral part of their work. Salton and McGill (1983) says that even though "most practitioners interested in the design and operations of actual retrieval systems are concerned only about applied computer science," that one must not fail to understand that IR has strong links to both computer science and "behavioral science, since retrieval systems are designed to aid human activities." The impetus for exploring the connection between IR and behavioral science was spurred by discussion on the concept of "relevance." Relevance is certainly one of the key pillars of IR, some might argue that it is *the* fundamental pillar of IR (Borlund, 2003). A strong association between relevance and user satisfaction has been accepted by most researchers. Relevance has been extremely problematic. Schamber (1994) lists 80 factors, suggested by the IR research literature (Cuadra and Katter, 1967; Rees and Schultz, 1967; Cooper, 1971b, 1973) that she studied for her article, that affect relevance. In fact, relevance has become so important that it has become an area of study in its own right (Schamber et al., 1990; Schamber, 1994). Methods for its evaluation have been the focus of many studies and much debate. Even at the present time, this debate continues. During this period, users' actions, thought processes, and characteristics were intensively examined and discussed. One of the key concepts driving research into the understanding of users' cognitive processes with respect to IR has been Belkin's Anomalous State of Knowledge (ASK) (Belkin et al., 1982).

The third period, which started around 1985 and is still continuing, has to do with the realization that information retrieval is inherently an interactive and dynamic process. Within the last two decades much has changed about users' information seeking processes. Technology has certainly been an important factor. During that time, there has been tremendous advances in computer accessibility for the masses, computational power, memory, storage (both in quantity and type), price decreases, computer networking, graphical interfaces, data transfer rates, to name a few. The end result of all of this is that the user during this period, and probably even more so today, is quite likely to own her or his own computer and do her or his own searching as contrasted to using an intermediary (e.g., a reference librarian) which was very common during the second period. Technology during this period greatly enabled the search process, which is highly interactive, dynamic, and iterative, to be often carried out, solely, by the user, in real-time instead of in batch. This process of searching for information, obtaining results, evaluating those results, possibly modifying the query in response to those results, and then using the refined query to search again, constitutes the body of a loop that may be repeated several times until the user achieves some degree of satisfaction or gives up. This is the notion of relevance feedback (Spink and Losee, 1996).

> Technologies such as CD-ROM and improved communication networks have widened the availability of computer-based retrieval systems. Others, such as full-text databases and hypertext and hypermedia systems, have enlarged our notion of what constitutes an information record in an information retrieval system. A paradigmatic shift has occurred in the research front, to user-centered from system-centered models. (Tague-Sutcliffe, 1992)

Tague-Sutcliffe, in the above quote from the 1992 update to her earlier paper (Tague, 1981), remarks that, in the interim between the publication of these two papers, the IR paradigm had shifted from system-oriented models to user-centered models. Relevance feedback is a key feature of many user-centered models.

2.5 IR Performance Evaluation and Test Collections

The experimental approach has been — by far — the predominant way to evaluate the retrieval performance of an IR system. To contrast, the approach that we chose in this dissertation was analytical – which could be viewed as being the direct opposite of the experimental approach. In this dissertation, the goal was *predictive*, that is, to determine how well an Information retrieval system, or some part of it, was likely to perform with some degree of confidence *before* the system processed a given query. The experimental approach, by its very nature, concerns itself with *retrospective* performance evaluation.

In this section, we continue by specifying the main elements in an IR performance evaluation and providing a brief history of early to current day IR test collection development. Following that, we provide a formal definition of a test collection and then discuss some trends that have been occurring in test collection development during the last decade. Next, we discuss some of the requirements for 'ideal' test collections; the overwhelming preponderance of these requirements come directly from the seminal article titled *Information Retrieval Test Collections* (Spärck Jones and van Rijsbergen, 1976). These requirements are mostly system-centered. Tague-Sutcliffe (1992) lists several additional requirements to make the user an integral part of the evaluation and collection development. Finally, we conclude this section by listing some criteria that help determine whether a specified test collection would be "good" or "bad" for a particular evaluation study. The test collections that were used in the research for this dissertation were discussed in Chapter 3 (Method), starting at the beginning there and continuing for several pages.

2.5.1 IR Performance Evaluation

The experimental approach uses a controlled experiment (i.e., laboratory-style methodology) to assess the performance of an IR system. An evaluation using this approach consists of these 3 elements: a set of queries, a set of documents, and relevance judgments for the relationship between each query and document in the reference collection. These three elements, taken together, have been referred to in the IR literature, at various times, as either a *benchmark collection*, a *reference collection*, a *test reference collection*, or simply a *test collection*. Out of these 4 alternatives, we elected to use the phrase "test collection" in this dissertation. A formal definition of a test collection appears in Section 2.5.2 on the next page.

Performance is measured by benchmarking. That is, the retrieval effectiveness of a system is evaluated on a given test collection. Figure 2.3 on the following page represents a prototypical experimental approach. Problems with benchmarking include the following: performance data is valid only for the environment under which the IR system is evaluated, building a benchmark corpus (i.e., the collection of documents) is a difficult task; using a benchmark without knowing, or respecting, the assumptions, constraints, and purposes that it was built for, can lead to misleading results.

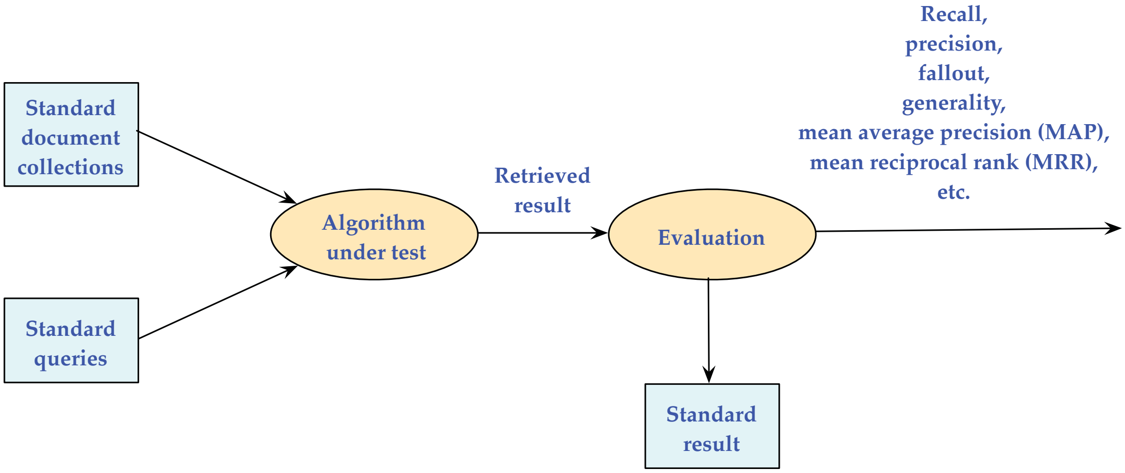

Standard document collections
Standard queries
Algorithm under test
Retrieved result
Evaluation
Recall, precision, fallout, generality, mean average precision (MAP), mean reciprocal rank (MRR), etc.
Standard result

Figure 2.3: The Prototypical Experimental Retrieval Performance Evaluation Schema. *Source:* Adapted from Mooney (2006).

2.5.2 A Formal Definition of a Test Collection

Definition. *An information retrieval model is a quadruple* $[\mathbf{D}, \mathbf{Q}, \mathcal{F}, R(q_i, d_j)]$ *where*

(1) $\mathbf{D}$ *is a set composed of logical views (or representations) for the documents in the collection.*

(2) $\mathbf{Q}$ *is a set composed of logical views (or representations) for the user information needs. Such representations are called queries.*

(3) $\mathcal{F}$ *is a framework for modeling document representations, queries, and their relationships.* [The most important relationship is the one that relates each query to its set of relevant documents.]

(4) $R(q_i, d_j)$ *is a ranking function which associates a real number with a query* $q_i \in \mathbf{Q}$ *and a document representation* $d_j \in \mathbf{D}$*. Such ranking defines an ordering among the documents with regard to the query* q_i*.* (Frakes and Baeza-Yates, 1992)

Using parts of the Frakes and Baeza-Yates definition, we define a *test collection* (TC) as a triple $[\mathbf{D}, \mathbf{Q}, \mathcal{F}]$ that has elements (1), (2), and (3) from that definition.

The purpose of the above definitions is to provide a formal basis for the notions of information retrieval model and test collection. These notions are particularly germane to the material and discussions that occur in Chapter 8 (Validation of the Formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ Measures), Chapter 9 (The ASL Performance Measure Variants and Empirical Document Rankings), and Chapter 10 (The ASL Measure and Three Frequently-Used Performance Measures).

2.5.3 Several Generations of Test Collections

Many of the early test collections are based on small test collections such as CACM (3,204 documents; 64 queries; 1.5 megabytes in size) (Fox, 1983), ISI (also known as CISI) (1,460 documents; 112 queries; 1.3 megabytes in size) (Fox, 1983), CRAN (also known as the Cranfield collection) (1,400 documents; 225 queries; 1.6 megabytes in size) (Cleverdon, 1997), MED (also known as MED1033) (1,033 documents; 30 queries; 1.1 megabytes in size) (Salton and Buckley, 1990), and TIME (425 documents; 83 queries; 1.5 megabytes in size) (`http://www.cs.utk.edu/~lsi/corpa.html` (last accessed on April 7, 2010)). Different researchers used different test collections and evaluation techniques.

The Cranfield tests on index language devices (Cleverdon, 1997) was a seminal event for information retrieval test collection development and performance evaluation. It established methodologies and procedures that are in use to this present day.

The first generation test collections – CRAN, CACM, CISI, and MED1033 – came about in the 1960s and 1970s and were characterized by their small size (using today's standards). At the time that they appeared though, their sizes were very reasonable once computer accessibility and cost in addition to storage availability and its costs were considered. The main emphasis during this time was on *ad hoc* queries. Quite often the document collections were not full-text. It was not an all unusual to find that a "document" consisted only of a title, an abstract, and some keywords.

The second generation test collections started in 1992 with the Text REtrieval Conference (TREC). They were orders of magnitude larger than those of the first generation and used pooling (Jones and van Rijsbergen, 1975) because the collections were so large. With the first generation test collections, it was possible — if one had enough perseverance and time — to find *all* of the relevant documents for each query. However, this was not feasible when the test collections started to become real large. Instead of judging each document, only the documents in the pool are judged. An implicit assumption, though, is that each relevant document is retrieved by an least one IR system. The TREC conferences have been of seminal importance to the areas of IR experimentation and evaluation. A brief history of it and its objectives appear in Section 2.5.5 on page 45. Test collection development during this area started to incorporate the notion of a "user."

The third generation test collections started at the end of the 1990s and continue to today. These collections are associated with well-known efforts such as the Amaryllis campaign (Landi et al., 1998), CLEF (Kluck, 2003), INEX (Lalmas, 2005; Lalmas and Tombros, 2007), NTCIR (Kando et al., 1999), and the ongoing development of new and expanded test collections for TREC (Voorhees and Harman, 2005).

The performance evaluation research discussed in this dissertation occurred during the third generation test collection period. Performance evaluation in this period, as in the prior two periods, has been mostly empirical and retrospective. The main contribution of this dissertation research was the creation of equations and procedures based on analytic and combinatoric concepts that could be used to predict, study, and obtain a better understanding of IR system performance, under certain circumstances, for a query or set of queries. The method of performance evaluation used in this dissertation contrasted sharply with the methods by which performance had been typically evaluated in IR during the third generation period. For example, at the TREC conferences, the performance evaluation process had been empirical and retrospective; the process used in this dissertation was analytical and predictive.

Whereas the first generation test collections were for *ad hoc* information needs, the second and third generation also developed collections for emerging and specialized retrieval areas. For example, the CLEF test collections focused on European languages and cross-language information retrieval; the INEX test collections were for the evaluation of XML retrieval in mostly *ad hoc* situations; and the NCTIR test collections were for East Asian languages and cross-language information retrieval.

2.5.4 Design Requirements for an Ideal Test Collection

The Spärck Jones and van Rijsbergen (1976) article appears to be the first instance in the IR literature of a comprehensive and detailed set of design requirements for test collections. In this article, explicit requirements are specified for test collections, *per se*, and also for their individual components (i.e., documents, requests, relevance judgments).

The motivation for these requirements are from their (and the IR research community's') "[e]xperience with the defects and limitations of past test collections . . . " (Spärck Jones and van Rijsbergen, 1976). One omnipresent problem was the small sizes, both in terms of documents and queries, of many of the collections being used in the research studies of that era.

Spärck Jones and van Rijsbergen (1976) explicitly states that these design requirements were developed from a non-exhaustive survey of approximately 30 text collections that had been used in various studies which had been reported in the research literature of that time. The requirements arose from 3 kinds of needs: purely formal needs related to statistical validity, needs related to the control of variables, and the need to be able to hopefully extrapolate experimental results to real-world systems.

Listed below are the specific requirements for test collections. The Spärck Jones and van Rijsbergen (1976) article specifies other sets of requirements for documents, requests, and relevance judgments but these are not given below because they are rather extensive and are detailed expansions of some of the material that appear as test collection requirements.

> The ideal collection(s) should also exhibit on the other hand variety in different respects, and on the other homogeneity. This is necessary both from an experimental point of view in that specific devices should be tested both for consistency and for discrimination; and from the point of view of representation, since test collections must reflect retrieval environments which are sometimes characterized by variety and sometimes by homogeneity. Thus we may say that from a material point of view, the ideal collection(s) should be
>
> 1. *various in content*: i.e. documents and requests should cover a range of subjects, e.g. science, social science, news, including subjects of difference specialization and hardness; and
> *homogeneous in content.*
> 2. *various in type*: i.e. documents should be of different kinds, e.g. popular, specialized, survey, etc., requests be e.g. broad, narrow; and
> *homogeneous in type.*
> 3. *various in source*: documents should cover a range of journals and journal types; and
> *homogeneous in source.*
> 4. *various in origin*: i.e. documents should represent authors of different origins and status, requests different users and different needs; and
> *homogeneous in origin.*
> 5. *range over time*: documents should be of different dates, and requests of different dates both for different users and the same user; and
> *coincide in time.*
> 6. *various in natural language*; and
> *homogeneous in natural language.* (Spärck Jones and van Rijsbergen, 1976)

2.5.5 Text REtrieval Conference (TREC)

TREC is an annual conference that originated from the TIPSTER program sponsored by the Defense Advanced Research Projects Agency (DARPA). It became an annual conference in 1992, co-sponsored by the National Institute of Standards and Technology

(NIST) and DARPA. Participants are given parts of a standard set of documents and topics (from which queries have to be derived) in different stages for training and testing. Participants submit the the values of various measures (e.g., recall, precision, mean average precision (MAP) (Voorhees, 2000), mean reciprocal rank (MRR) (Voorhees and Tice, 1999)) for the final document and query corpus and present their results at the conference.

The motivations for starting TREC were varied. The passages from Hersh (2003) and Buckley and Voorhees (2005) below provide some insight.

> One of the motivations for starting TREC was the observation that much IR evaluation research (prior to the early 1990s) was done on small test collections that were not representative of real-world databases. Furthermore, some companies had developed their own large databases for evaluation but were unwilling to share them with other researchers. TREC was therefore designed to serve as a means to increase communication among academic, industrial, and governmental IR researchers. Although the results were presented in a way that allowed comparison of different systems, conference organizers advocated that the forum not be a "competition" but instead a means to share ideas and techniques for successful IR. In fact, participants are required to sign an agreement not to use results of the conference in advertisements and other public materials .... (Hersh, 2003)

> In the three to four years immediately preceding TREC-1 [the *first* TREC conference], test collection evaluation as seen in published papers had become increasingly chaotic. Computing resources had become cheap enough so that many more groups could perform retrieval experiments, but the groups did not agree on how to evaluate those experiments. Papers reported scores for only the authors' preferred measure, when each of the following was preferred by someone [...]: precision at ten documents, recall measures, utility, full recall-precision curves, three-point averages from the recall-precision curves, ten-point averages, and eleven-point averages. Even when papers reported what they called the same measure — for example, a three-point average — the implementation of the measure often differed [...]. Thus, it was unusual that the results presented in any two papers could legitimately be compared to each other, despite having used the same test collections. This was a major problem when trying to learn from papers of the era. The reader was never quite sure whether a single system evaluation comparison showed a poor system becoming mediocre or a good system actually demonstrating a technique that was generally useful. (Buckley and Voorhees, 2005)

The objectives of TREC were to provide a common ground (e.g., same set of queries and documents, same evaluation method) for comparing different IR techniques, to encourage participation from industry and academia, and to foster the development of new evaluation techniques, particularly for new applications.

The primary advantages of TREC were the large size of the test collections, the use of full-text (as contrasted with abstracts), the queries and relevance judgments provided, the continuous development with support from the U. S. Government, and that the "careful attention paid to appropriate design criteria will allow unanticipated use in future experiments to be successful" (Sparck Jones, 2005). Elaborating on this, Sparck Jones writes:

> The TREC collections have been formed with care, to obtain realistic document files and requests as well as extensive relevance assessments. Moreover, with several different collections and broadly based relevance pools, the results obtained should be free from hidden biases and usefully general or generalizable. (Sparck Jones, 2005)

When comparing the results of IR systems, "[a]n important element of TREC is to provide a common evaluation for the systems. TREC reports a variety of recall- and precision-based evaluation measures [...]" (Harman, 2005). Four basic types of evaluation measures were in use at the TREC conferences: summary table statistics, recall-precision averages, document level averages, and average precision histograms (Baeza-Yates and Ribeiro-Neto, 1999).

2.6 Constructing Single Term Queries

This research used single term queries. A particularly vexing problem with respect to this research was that the queries in all of the candidate test collections (described starting on page 62 in Section 3.1.1) had multiple (as contrasted to single) terms. The question that quickly appeared was "How do we obtain single term queries for the set of documents in an arbitrary test collection?" There were several ways to go about this, none of them were particularly appealing. But, when the query had to be single term, there just were not many good choices for mapping, or distilling, a multiple term query into a single term one that had a strong semantic relationship with the intended meaning of the query.

One possible way would be to replace each multiple term query by a randomly chosen term that appeared in the original query. The main advantage of this approach was that the relevance judgments that came along with the test collection could still be used. Some rather obvious problems with this were that the random term could be a function word (e.g., the, a, of) or it could be a content word that had a minimal relationship to the spirit of the query. For example, suppose the query was related to some aspect of malaria-eradication efforts by the United Nations, the term *bookmark* appeared in the query, and this term was chosen to be the single term representative of this query. The odds were that this term would not a a good representative for the topic that this query addresses.

Another possible way might be to select the most frequently occurring content term in the query as the representative. One problem with this way was that this term may have a very loose semantic relationship with the meaning of the query. Another problem might be that there were several content terms that have a frequency that is higher than that of any of the others. Which one should be chosen?

Still another possible way involved breaking an n-term query into n one-term queries, then choosing from among these one-term queries the one that had the best semantic relationship to the multiple term query. If more than one "best" query was found, then one of them could be randomly chosen as the single term representative.

It is very important to the scientific credibility of any multiple- to single-term distillation technique that the resultant term be as strongly related, as is technically possible, given the very tight constraint (i.e., reducing many terms to just one) that researchers, or computer algorithms, often work with, to the set of relevant documents that is associated with the query. This means that a researcher, or software algorithm, has to be careful about the manner in which this representative is chosen. It is imperative that whatever manner is chosen be scientifically defensible.

A much better way to effect this single term representation is a strategy that generates a synthetic single term query for each query in the collection in such a way that the term best represents the query with respect to its relevance set, and the relevance sets of all the other other queries, that comprise the query set for the collection. The generated query for a query and relevance set combination could be viewed as a one-term summarization, or distillation, of those in the relevance set documents. Note that the best term may not be one that is in the query. A mechanism to select such a term, using machinery from language models and information theory, is described in the next section.

2.7 Language Models and Relative Entropy

The language model (LM) approach (Ponte and Croft, 1998) to document ranking is a probabilistic approach. It differs significantly, however, from the classic probabilistic model in that it does not attempt to group documents into relevant and non-relevant categories. The language model approach ranks a document according to how likely its model would produce the query. In this approach, each document and each query has its own language model associated with it. A language model for a query-document pair is a probability distribution over the terms comprising that query-document pair. Since the number of terms is finite and the distribution is discrete, a language model is a probability mass function. That means, among other things, we can use information theory concepts (Cover and Thomas, 2006; Jones and Jones, 2000; Jelinek, 1997; Luenberger, 2006) to compare two language models whose elements range over the same domain. The mechanism described here is used in Section 3.2.1 to map a multiple term query to a single term query.

The interest in language models for this research was limited to just using some of the theory associated with them and information theory to help construct single term queries. At this point in the discussion, one might naturally ask questions similar to “What was so special about the language model-based approach to picking a single term representative? Would other approaches have worked just as well, or better? How about just randomly choosing one of the content words in the query as its single term representative – it certainly would be a lot simpler than this LM-based approach?" The answers to the first two questions are discussed below. The answer to the third question is in Section 2.6.

In this dissertation, the primary reasons for selecting this LM approach were that information retrieval language models seek to model the query generation process, have a very sound theoretical grounding, and are frequently used in various studies to help with automatic query generation (Lafferty and Zhai, 2001; Berger and Lafferty, 1999). Another important reason was that this research may be extended one day to handle multiple term queries. An attractive feature of choosing the particular distillation approach used in this dissertation was that the approach could be easily modified to generate two-term queries, three-term queries, or queries with a somewhat arbitrary number of terms. On a historical note, Lafferty and Zhai remark:

> Interestingly, the very first probabilistic model for information retrieval, namely the Probabilistic Indexing model of Maron and Kuhns ... is, in fact, based on the idea of "query generation." Conceptually, the model intends to infer the probability that a document is relevant to to a query based on the probability that a user who likes the document would have used this query. However, the formal derivation given in ... appears to be restricted to queries with only a single term. (Lafferty and Zhai, 2003)

One technique to selecting a single term query, from one that had multiple terms, would be to randomly select a term from the original query, but doing this came with its own share of problems. The research in this dissertation did not have to be concerned about these problems because there had been a good amount of research based on the use of the information-theoretic concept of divergence to facilitate term selection in areas such as automatic query expansion (Cai et al., 2001; Cai and van Rijsbergen, 2004), model-based feedback (Zhai and Lafferty, 2001), and the generation of queries of various qualities for blind relevance feedback (Jordan, 2005; Jordan et al., 2006).

In particular, we were interested later in making use of the scoring portion (the part after the Σ (i.e., summation)) symbol of the *relative entropy* (also known as *Kullback-Leibler divergence*) formula (Equation 2.7.1) from information theory to determine which vocabulary term is the least significant contributor to the divergence between two language models. In the notation below, RE is the abbreviation for relative entropy; KL is the shorthand for Kullback-Leibler divergence; d denotes a document; q denotes a query; w denotes a word; M_q denotes the language model for query q; M_d denotes the language model for document d; $P(w|M_q)$ denotes the probability that word w occurs in M_q; and $P(w|M_d)$ denotes the probability that word w occurs in M_d, and log denotes the natural logarithm.

Note that the notation for probability that is used later in this particular section is different from the notation that is used in the remainder of this dissertation. In this section, namely, Section 2.7, the notation used is the same one that is in the Manning et al. (2008) block quote that appears at the end of this section. The reason for using the same notation throughout this section, both inside and outside of the block quote, is to minimize notational confusion. $P(a)$, in this section, denotes the probability of event a and $P(a|b)$ denotes the probability of event a given that event b has occurred. Outside of this section, the author prefers the use of $\Pr(a)$ and $\Pr(a|b)$, respectively.

The general equation for Kullback-Leibler divergence is

$$\mathrm{RE}(d; q) = \mathrm{KL}(d \parallel q) = \sum_{w} P(w|M_q) \log \frac{P(w|M_q)}{P(w|M_d)}. \tag{2.7.1}$$

If we slightly alter Equation 2.7.1, by replacing the symbol for word w with the symbol t for a term over a vocabulary V, we obtain Equation 2.7.2, a form of the equation that is more specific to information retrieval.

$$\mathrm{RE}(d; q) = KL(d \parallel q) = \sum_{t \in V} P(t|M_q) \log \frac{P(t|M_q)}{P(t|M_d)}. \tag{2.7.2}$$

Relative entropy (Cover and Thomas, 2006; Jones and Jones, 2000; Jelinek, 1997) measures how dissimilar two probability mass functions are. Smaller values indicate greater similarity; larger values indicate greater dissimilarity.

How do we obtain $P(t|M_d)$? We typically have to estimate it. A popular way to do that is via the use of a technique known as *maximum likelihood estimation* (MLE) (Law, 2006; Rose and Smith, 2002; Terrell, 1999). MLE is a statistical method for making inferences about population parameters (e.g., mean, variance) of the underlying probability distribution from sample data. The values that are estimated for the parameters are those that are "most likely" given the sample data; i.e., they have the greatest probability (likelihood) of obtaining the sample data.

According to Manning et al. (2008), "[t]he probability of producing the query given the LM M_d of document d using maximum likelihood estimation (MLE) and given the unigram [bag of words] assumption is:

$$\widehat{P}(q|M_d) = \prod_{t \in q} \widehat{P}_{\text{mle}}(t|M_d) = \prod_{t \in q} \frac{\text{tf}_{t,d}}{\text{L}_d}, \tag{2.7.3}$$

where M_d is the LM of document d, $\text{tf}_{t,d}$ is the (raw) term frequency of term t in document d, and L_d is the number of tokens in document d." The symbol Π denotes multiplication.

A quick inspection of Equation 2.7.3 readily reveals a possible problem: if the query has a term t that does not appear in document d, then the MLE probability is 0. What do we do when a term t is present in the query but is not in the document model? The passage below provides some insight about this issue.

> The classic problem with using LMs is one of estimation (the ˆ [i.e., caret] symbol on the Ps is used above [in Equation 2.7.3] to stress that the model is estimated): Terms appear very sparsely in documents. In particular, some words will not have appeared in the document at all, but are possible words for the information need, which the user may have used in the query. If we estimate $\hat{P}(t|M_d) = 0$ for a term missing from a document d, then we get a strict conjunctive semantics: Documents will only give a query nonzero probability if all of the query terms appear in the document. Zero probabilities are clearly a problem in other uses of LMs, such as

when predicting the next word in a speech recognition application, because many words will be sparsely represented in the training data. It may seem rather less clear whether this is problematic in an IR application. This could be thought of as a human-computer interface issue: Vector space systems have generally preferred more lenient matching, although recent web search developments have tended more in the direction of doing searches with such conjunctive semantics. Regardless of the approach here, there is a more general problem of estimation: Occurring words are also poorly estimated; in particular, the probability of words occurring once in the document is normally overestimated, because their one occurrence was partly by chance. The answer to this . . . is smoothing. But as people have come to understand the LM approach better, it has become apparent that the role of smoothing in this model is not only to avoid zero probabilities. The smoothing of terms actually implements major parts of the term weighting component . . . . It is not just that an unsmoothed model has conjunctive semantics; an unsmoothed model works badly because it lacks parts of the term weighting component.

Thus, we need to smooth probabilities in our document LMs to discount nonzero probabilities and to give some probability mass to unseen words. There's a wide space of approaches to smoothing probability distributions to deal with this problem. In Section . . . , we already discussed adding a number(1, 1/2, or a small α) to the observed counts and renormalizing to give a probability distribution. In this section, we mention a couple of other smoothing methods that involve combining observed counts with a more general reference probability distribution. The general approach is that a nonoccurring term should be possible in a query, but its probability should be somewhat close to but no more likely than would be expected by chance from the whole collection. The general approach is that a non-occurring term is possible in a query, but no more likely than would be expected by chance from the whole collection. That is, if $\mathrm{tf}_{t,d} = 0$ then

$$\widehat{P}(t|M_d) \leq \mathrm{cf}_t/\mathrm{T}$$

where cf_t is the raw count of the term in the collection, and T is the raw size (number of tokens) of the entire collection. A simple idea that works well in practice is to use a mixture between a document-specic multinomial distribution and a multinomial distribution estimated from the entire collection:

$$\widehat{P}(t|d) = \lambda \widehat{P}_{\mathrm{mle}}(t|M_d) + (1-\lambda)\widehat{P}_{\mathrm{mle}}(t|M_c)$$

where $0 < \lambda < 1$ and M_c is a language model built from the entire document collection. This mixes the probability from the document with the general collection frequency of the word. Such a model is referred to as a *linear interpolation* LM. Correctly setting λ is important to the good performance of this model.

. . .

The extent of smoothing in [this model] is controlled by the λ [parameter]: a small value of λ . . . means more smoothing. This parameter can be tuned to optimize performance using a line search (or, for the linear interpolation model, by other methods, such as the expectation maximimization algorithm; . . .). The value need

> not be a constant. One approach is to make the value a function of the query size. This is useful because a small amount of smoothing (a "conjunctive-like" search) is more suitable for short queries, whereas a lot of smoothing is more suitable for long queries.
>
> To summarize, the retrieval ranking for a query q under the basic LM for IR we have been considering is given by
>
> $$P(d|q) \propto P(d) \prod_{t \in q} ((1-\lambda)P(t|M_c) + \lambda P(t|M_d)).$$
>
> The equation captures the probability that the document that the user had in mind was in fact d. (Manning et al., 2008)

The expression above of the form $a \propto b$ denotes that a is proportional to b.

2.8 Statistical Significance in Query System Performance

The purpose of much of the research in this dissertation was to show that the combinatoric results were similar to the empirical results — it was not to obtain the best performance results. Among the main objects of interest were ranked data and the performance of various ranking methods with respect to this data. A central question was "How can it be determined that the combinatoric results are statistically similar to the empirical results?" A complicating matter was that the ranks are ordinal and the data typically did not fit any known distribution; therefore, the use of parametric statistics was generally inappropriate (much more about that topic is discussed later in this section). So, the question became "Given the nature of the data used in *this* research and its research goals, how can statistical significance be determined? What are the appropriate significance tests to use for *this* research?"

The Kolmogorov-Smirnov (K-S) goodness-of-fit test (Conover, 1999) and the Mann-Whitney test (also known as the Wilcoxon signed ranks test) (Conover, 1999) were the two main significance tests used in this research. The K-S test was used for part of RQ

#1 (determining the characteristics of a combinatoric-based ASL performance measure) and both tests were used for RQ #2 (determining how well the results predicted by a combinatoric-based ASL matches up with the results obtained from actual document rankings). The example in Section 3.4 provides more information about the context in which this research employed the Kolmogorov-Smirnov test. The remainder of this section discusses general statistical significance issues in IR performance research.

Van Rijsbergen (1979) states that "[o]nce we have our retrieval effectiveness figures we may wish to establish that the difference in effectiveness under two conditions is statistically significant. It is precisely for this purpose that many statistical tests have been designed. Unfortunately, ... there are no known statistical tests applicable to IR. This may sound like a counsel of defeat but let me hasten to add that it is possible to select a test which violates only a few of the assumptions it makes."

> The use in IR experiments of formal statistical methods such as significance tests has been relatively unusual. This gap has to do in part with the difficulty of establishing the validity of particular tests or even of defining a suitable framework for such tests (IR experimental data is notoriously difficult to pin down in any neat statistical model). ... One problem that needs to be addressed when deciding on a statistical significance test, is what (if any) assumptions can be made about the shapes of the distributions. Many tests depend on strong assumptions about these shapes. Unfortunately, IR is notoriously difficult to pin down in this respect. Of course, the actual distribution will depend on which particular variable is being measured as well as the circumstances of measurement; but many authors have pointed to the difficulty of justifying any parametric assumptions. We are therefore lead towards nonparametric tests (Siegel, 1956). (Robertson, 1990)

An earlier article (Robertson, 1981) discusses some of the difficulties.

Harter and Hert (1997) remarks that "[t]he role of significance testing and other statistical issues related to retrieval evaluation have not been treated to any great extent in the retrieval literature. In part this has been because the assumptions underlying statistical treatment (independence, random sampling, assumptions of normality and the like) are rarely met by Cranfield instruments ...."

One implication of the three paragraphs above is that it may be hard to use parametric tests (e.g., t-test, F-test, analysis-of-variance tests) for significance testing in information retrieval research. Many of the hypothesis-testing procedures used in science and engineering for parametric statistics are based on the assumption that the random samples are selected from normal populations. Many of these tests are still reliable when there are slight deviations from normality, especially when the sample size is sufficiently large. If parametric tests are used, in general, one or more of the statistical assumptions that they are based on may have been violated and, depending on the degree of violation and the robustness of the test, the p-value may have a sizable amount of error. Walpole (2002) remarks that "this is particularly true for the t-test and the F-test." Depending on the robustness of the technique and other factors, this may or may not be a problem. If it does turn out to be a problem, then researchers often have to resort to using nonparametric (i.e., distribution-free) statistical methods. The primary downside of non-parametric tests is that "they do not utilize all of the information provided by the sample, and thus a nonparametric test will be less efficient than the corresponding parametric test. Consequently, to achieve the same power, a nonparametric test will require a larger sample size than will the corresponding parametric test" (Walpole, 2002).

Van Rijsbergen states, with respect to significance testing in IR, that "[o]n the face of it non-parametric tests might provide the answer"(van Rijsbergen, 1979). He mentions one particular case where there is a single set of queries that is used in different retrieval environments:

> Therefore, without questioning whether we have random samples, it is clear that the sample under condition a is related to the sample under condition b. When in this situation a common test to use has been the Wilcoxon Matched-Pairs test. Unfortunately again some important assumptions are not met. The test is done on the difference $D_i = Z_a(Q_i) - Z_b(Q_i)$, but it is assumed that D_i is continuous and that it is derived from a symmetric distribution, neither of which is normally met in IR data.
>
> It seems therefore that some of the more sophisticated statistical tests are inappropriate. There is, however, one simple test which makes very few assumptions and which can be used providing its limitations are noted. This one is known in the literature as the sign test (Siegel[29], page 68 and Conover[30], page 121). It is applicable in the case of related samples. It makes no assumptions about the form of the underlying distribution. It does, however, assume that the data are derived from

> a continuous variable and that the $Z(Q_i)$ are statistically independent. These two conditions are unlikely to be met in a retrieval experiment. Nevertheless, given that some of the conditions are not met, it can be used conservatively. (van Rijsbergen, 1979)

One particular arena of applicability for nonparametric tests in IR research has to do with the fact that much of results evaluation in that area involves the comparison of ranked (i.e., ordinal scale) results. Parametric tests are ill-equipped to deal with these as the analysis of this ordinal data involves an *analysis of ranks*. This kind of analysis can, however, be very naturally handled by their nonparametric counterparts. Some IR literature examples of, or references to, the use of non-parametric tests in IR are the following: the Kolmogorov-Smirnov one-sample test for goodness-of-fit (Moon, 1993), the Wilcoxon-Mann-Whitney test (Keen, 1992), the sign test (Downie et al., 2005), McNemar's test (Downie et al., 2005), and the Wilcoxon signed ranks test (Downie et al., 2005). These are just a sampling of the tests that were available for possible use in this dissertation. Generally, the tests that are used in a particular situation depend very much on the characteristics of the situation and the researcher's goals.

2.9 Significant Sample Sizes for Document Collections and Queries

Robertson is the author of one of the early articles (Robertson, 1981) that solely addresses methodological issues, in general, and sample sizes for document collections and queries, in particular. Robertson (1981) states that, for the variable(s) – which may be a measure (such as recall or precision), a cost, or some other entity – of interest in an experiment or study, the acquisition of an adequate collection of documents is generally not a problem; however, obtaining a sufficient number of queries can be very problematic. Additionally, Robertson (1981) states that the problem is not so much with the number of queries, but in obtaining a representative sample of them. Also, Robertson (1981) states that "'trapping' the queries at an appropriate moment of their existence and obtaining the necessary co-operation of the requesters, is by no means a trivial task." Due to that difficult task, many of the early studies only used a few tens, rather than a few hundreds, of queries and, thereby, had questionable validity. Another problem with queries, almost independent of the measure of interest, is that they typically have a wide variation for that measure whereas the difference between the systems that are being compared can be relatively small. Robertson's article goes on to state that time can be a problem with document collections. Two of the examples given have to do with a collection's subjects changing over time or the proportions of the documents for each subject varying over time.

Robertson (1990) discusses the problem of determining an adequate sample size for comparing two IR systems that have separate (i.e., independent) samples of requests. Many Cranfield-style experiments use a "matched-pair" or "repeated measurement" design. The problem with that, especially with online, interactive, and iterative requests is that once the user makes a request and responds to the results of that request, she or he no longer has the same Anomalous State of Knowledge (ASK) (Belkin et al., 1982) as before. This problem is also known as the *learning effect* (Harada et al., 2004). Using a non-matched-pair design is a way to counteract these interaction problems. With the assumption that the experimental design has independent samples, this paper provides guidance for determining the sample size calculation for various distributional assumptions. There are sample size calculations for rectangular distributions, trapezium distributions, normal distributions, exponential distributions, normal distributions with the t-test, and binary distributions with the chi-squared test. Near the conclusion of this paper, Robertson pointed out several limitations of his study: only two IR systems were involved; its focus was "on a small number of somewhat artificial distributions" (Robertson, 1990); the distributions were mainly continuous, "[r]eal-life distributions tend to be a lot messier, and in this respect the results are indicative only" (Robertson, 1990); and it only provided "for tests of 50% power [...]. Results requiring higher power would involve [even larger] samples [...]" (Robertson, 1990).

From the earlier paragraphs in this section, it was stated that two of the major methodological issues that have bedeviled information retrieval researchers had been how to obtain a representative sample of queries and how to determine an adequate sample size for research projects. Generally, there has been a scarcity of literature that solely focused on research methodology for IR system evaluation and that provided some guidance on those and other issues. The guidance that was available for issues such as those discussed above was typically buried in the research methods sections of individual journal articles and was not treated in a comprehensive, cohesive, and uniform way. This created difficulty in getting started for many scholars new to IR research. Robertson (2001) says "[t]he methods and techniques associated with the evaluation of IR systems ... tend to be described in the methodology sections of research reports and papers. It is unusual to see papers or monographs devoted to methodology *per se*."

One notable exception to Robertson's assertion above is the collection of papers edited by Karen Sparck Jones. That collection (Jones, 1981) is slightly over a quarter century old now (and somewhat outdated) but "must [still] be regarded as the classic source in the field" (Robertson, 2001).

2.10 Summary

This chapter introduced the ASL measure, the normalized average search length $\mathcal{A}$, the notion of a ranking (i.e., a sequence of ordered documents), and the three alternative measures (i.e., ESL, MRR, MZE) that the performance of the ASL measure is compared with in Chapter 10 (The ASL Measure and Three Frequently-Used Performance Measures).

This chapter also contained discussions on several other topics: the mathematical machinery (i.e., notation, proofs, probability theory and models, the query-document model, combinatorial generation and enumeration algorithms) that were used in subsequent chapters; term and query operations (i.e., lexical analysis, stoplists, stemming); an historical view of information retrieval research; IR performance and test collections; transforming multiple term queries to single term queries; statistical significance; and significant sample sizes for document collections and queries.

Chapter 3

Method

The general method for carrying out this research consisted of steps to obtain the Cystic Fibrosis (CF) test collection (Shaw et al., 1991) and other instruments that were used to generate the input data for this research. Afterwards, a new test collection namely, CF$'$, was created from the original CF test collection. This new collection was a slightly modified version of the original collection. The purpose for creating it was to change the data into a form more suited for the needs of this research.

Additionally, synthetic datasets and random sets of queries were created to help with the testing and validation of equations that were developed for each of the 3 research questions. Test data generation, and verification that the analytically-determined results matched the empirically-determined results, are discussed in detail in Chapter 8 (Validation of the Formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ Measures).

3.1 Instruments

This research used 5 instruments, the main ones were the CF test collection and several synthetic datasets and sets of random queries. The other instruments were the PubMed stopword list, a lexical analyzer, and the Porter stemmer. Each of these is discussed below.

3.1.1 The Cystic Fibrosis Test Collection

The Cystic Fibrosis test collection (Shaw et al., 1991) contains 1239 documents and 100 queries related to medicine. Each document has a document identifier (i.e., *did*) associated with it and each query has a query identifier (i.e., *qid*) associated with it. These identifiers are unique positive integer values with respect to their document and query domains. Each of the queries has a *relevance set* (*relset*) associated with it. The relset of a query identifies all the documents that are relevant to that query and the set of relevance judgments for each relevant document. A relevance judgment is one of three values: highly relevant, marginally relevant, not relevant. Each set of relevance judgments has cardinality 4 because each query-document pair was judged by 4 individuals. Operationally, a relset for a query q (identified by a qid) is a set that contains the dids of the relevant documents for that query. Associated with each did is a set of 4 values, with each of the relevance judgments encoded as either 0 (not relevant), 1 (marginally relevant), or 2 (highly relevant). More information about the CF test collection can be found in Section 3.2.1. The CF test collection is one of several collections that were used in this research and can be obtained from links on the `http://people.ischool.berkeley.edu/~hearst/irbook/cfc.html` (last accessed on April 7, 2010) Web page. The major advantage that this collection possessed was that it was free and readily available for download from several World Wide Web sites. The main disadvantage, with respect to the single term queries studied in this research, was that the relevance judgments in this collection were based on multiple term queries. In order to use any of the queries for this research, each of the multiple term queries needed to be represented by a single term query. Another disadvantage was that the highest level "document" information that this collection contained were abstracts.

3.1.2 Synthetic Datasets and Random Sets of Queries

The CF test collection was valuable for some of the small scale testing that occurred during the research that this dissertation undertook. However, it did not have a sufficient number of queries, number of documents, or queries with certain characteristics, that were needed in the latter chapters of this dissertation for result validation at the .05 and .01 significance levels. Synthetic documents and queries were generated to obtain the necessary numbers and varieties of entities with the desired characteristics.

3.1.3 PubMed stopword list

This is a general list of words that PubMed found to have little value in describing the information content of the documents in its collection. These words are known as *stopwords* (Baeza-Yates and Ribeiro-Neto, 1999; Grossman and Frieder, 2004; Meadow et al., 2007; Manning et al., 2008) in the information retrieval (IR) literature.

The PubMed stopword list was used to eliminate words that had low discrimination power, with respect to that domain, from the documents and queries in the Cystic Fibrosis test collection. The U. S. National Library of Medicine's official list of stop-words were obtained via this URL: `http://www.ncbi.nlm.nih.gov` (last accessed on April 7, 2010). The official list, as of this date, appears in Table 3.1 on the following page.

3.1.4 Lexical Analyzer

The particular generator used is the one that appears in Figure 7.8 of Frakes and Baeza-Yates (1992). Its source code was downloaded from this URL: `http://www.dcc.uchile.cl/~rbaeza/iradsbook/irbook.html` (last accessed on April 7, 2010).

Table 3.1: The PubMed Stopword List.

	Stopwords
A	a, about, again, all, almost, also, although, always, among, an, and, another, any, are, as, at
B	be, because, been, before, being, between, both, but, by
C	can, could
D	did, do, does, done, due, during
E	each, either, enough, especially, etc
F	for, found, from, further
H	had, has, have, having, here, how, however
I	i, if, in, into, is, it, its, itself
J	just
K	kg, km
M	made, mainly, make, may, mg, might, ml, mm, most, mostly, must
N	nearly, neither, no, nor
O	obtained, of, often, on, our, overall
P	perhaps, pmid
Q	quite
R	rather, really, regarding
S	seem, seen, several, should, show, showed, shown, shows, significantly, since, so, some, such
T	than, that, the, their, theirs, them, then, there, therefore, these, they, this, those, through, thus, to
U	upon, use, used, using
V	various, very
W	was, we, were, what, when, which, while, with, within, without, would

3.1.5 Porter stemmer

The implementation used is the algorithm that appears at the end of Chapter 8 of Frakes and Baeza-Yates (1992). Its source code was downloaded from this URL: `http://www.dcc.uchile.cl/~rbaeza/iradsbook/irbook.html` (last accessed on April 7, 2010).

3.2 Procedure

This section discusses the adaptation of the Cystic Fibrosis test collection, the creation of the synthetic datasets, and the creation of the sets of random queries.

3.2.1 Adapt the Cystic Fibrosis test collection

This involved preprocessing the Cystic Fibrosis test collection to get it into a form more amenable for performing this dissertation research. The details of this procedure are discussed in the next few paragraphs.

Assume that an instance of the CF test collection was represented by $[D_{\text{CF}}, Q_{\text{CF}}, J_{\text{CF}}]$. Notationally, let D_{tc}, Q_{tc}, and J_{tc} represent the sets of documents, queries, and relevance judgments, respectively, for a test collection tc. A query $q = q_1q_2...q_m$ has $m \geq 0$ terms, a document $d = d_1d_2...d_n$ has $n \geq 0$ terms. Each document or query may have a different number of terms than many others of its kind. A document or query was considered *trivial* if it had zero terms and *nontrivial* otherwise. Generally, it was expected that queries consisted of possibly several terms and documents to consist of many more. However, it was possible, though highly unlikely, that after stopword removal, a degenerate situation could occur where a document, or a query, would have no remaining terms. It was also possible that trivial documents or queries could result in some circumstances. This research assumed that all documents and queries were nontrivial, both before and after stopword elimination. Each query-document pair in a collection assumably had a unique numeric identifier and was represented by an ordered pair (i.e., $<id, entity>$) where *id* was the query-document pair identifier and *entity* was the bag of query-document pair terms. Each document in a collection had exactly one identifier associated with it; likewise, each identifier was associated with exactly one document. Stated more succinctly, there was a bijection (i.e., one-to-one correspondence) between documents and their identifiers and there also existed a bijection between queries and their identifiers. Without loss of generality, let the numeric identifiers for the documents in D_{tc} be integers that ranged from 1 to $|D_{\text{tc}}|$, inclusive, and, for queries, ranged from 1 to $|Q_{\text{tc}}|$, inclusive. The expression $|a|$ denoted, in general, the cardinality of set a (or bag a).

A relevance judgment was an ordered triple $<qid, did, rj>$ which represented the fact that the query with identifier *qid* and the document with identifier *did* had a joint relevance judgment value of rj. The relevance judgment value can take various forms, depending on the collection, but, in this research, it was assumed that the form did not vary within a collection. For example, the relevance judgment value was represented by an ordered quadruple of natural numbers in the closed interval $[0, 2]$ for the CF collection. More formally, a CF relevance judgment consisted of judgments by four distinct individuals and had the structure $<rj_1, rj_2, rj_3, rj_4>$ where $rj_{i\in\{1,2,3,4\}} \in \{0, 1, 2\}$ and each individual judgment had a value of 0 (not relevant), 1 (marginally relevant), or 2 (highly relevant).

The main ideas in the several paragraphs above can be summarized by stating that, for test collection tc:

Q_{tc} was a set of $<qid, bag\_of\_query\_terms>$ pairs;

D_{tc} was a set of $<did, bag\_of\_document\_terms>$ pairs; and

J_{tc} was a set of $<qid, did, relevance\_judgment(s)>$ triples.

Define access() as the *accessor function* for an n-tuple of the form $<v_1, v_2, ..., v_n>$.

Informally, an accessor function references a specified element of an n-tuple. Depending on context, the accessor can be used to either retrieve a value from the n-tuple or to change one of its values. In a retrieval context, the expression

$$\text{access}(<v_1, v_2, \ldots, v_n>,\ i)$$

yields v_i for $i \in \{1, 2, ..., n\}$, $n > 0$; it is undefined, otherwise.

In order to change the CF collection into the form needed by this research, a series of transformations were applied to its elements. These transformations yielded the transformed collections $\text{CF}' = [D_{\text{CF}'}, Q_{\text{CF}'}, J_{\text{CF}'}]$.

D_{CF} and Q_{CF} were transformed into $D_{\text{CF}'}$ and $Q_{\text{CF}'}$ by, conceptually, first using their associated stopword lists to remove any terms that are stopwords and placing the results into $D_{\text{CF}'}$ and $Q_{\text{CF}'}$, respectively. Stemming was then applied to both $D_{\text{CF}'}$ and $Q_{\text{CF}'}$. Finally, J_{CF} was transformed into $J_{\text{CF}'}$ via a simple mapping process.

Specify the operational definition of relevance

This was very important because the relevance judgment for a query-document pair was not simply a relevant, or non-relevant, value in the CF test collection. In particular, each query-document pair had four relevance judgments associated with it (Shaw et al., 1991; Shaw, 1995). Each judgment was one of three values (i.e., highly relevant, marginally relevant, not relevant). With these judgments, relevance could be defined in various ways. Two of the many possible ways that a document could be relevant to a query were if (1) at least one of its four judgments for that query was ‘highly relevant’ or ‘marginally relevant’ or (2) a document was relevant if at least one judgment for that query was ‘highly relevant’ and the majority of the remaining judgments were either 'highly relevant' or 'marginally relevant'. In this research, a document was relevant to a query for the CF test collection when the condition that was denoted by Way 1 was true.

Create the CF′ test collection

This process consisted of first eliminating the stopwords from the queries and the documents. Next, a new set of relevance judgments associations was built by visiting each of the original associations and mapping the four relevance judgments there into a single Y (relevant) or N (not-relevant) judgment. More details about the steps can be found in Appendix A.1, starting on page 515.

Select the best single term description of each query in the CF′ test collection

This process consisted of using language model theory (Ponte and Croft, 1998; Lafferty and Zhai, 2001) to determine the best single term to represent a multiple term query (Jordan et al., 2006). More details about this can be found in Appendix A.2, starting on page 517.

Create a composite query for each single term description that maps to multiple queries

Unfortunately, the process of distilling a multiple term query into a single term query was not guaranteed to produce a unique single term query for each of the original queries once the entire collection of queries was taken into account. This process resulted in a CF′ test collection that had only 74 unique single term queries out of a possible maximum of 100 unique single term queries. Fifty-eight of these single term representations occurred 1 time, eleven terms occurred 2 times, one term occurred 3 times, three terms occurred 4 times, and one term occurred 5 times.

The question of how the query set was going to be represented immediately arose. Should the query set only contain the 58 queries that had frequencies of 1? If the answer was negative, then how should the 16 query terms that had frequencies of two or greater be handled? Two of the possibilities for representing the queries were: (1) use only the 58 queries that corresponded to the query terms that occurred exactly once or (2) augment these 58 with composite queries for the 16 query terms with frequencies that were two or greater.

There were two main alternatives for creating the relevance set (of document identifiers) for a composite query. Conceptually, the first alternative was the creation of a query-term specific relevance set that consisted of the *union* of the relevance sets that were associated with each query in the in the CF$'$ test collection that was being described by that term. The other alternative was the creation of a query-term specific relevance set that consisted of the *intersection* of these relevance sets. Put another way, a document identifier was in the relevance set for a composite query for term t only if it was a member of the relevance set for at least one of the CF$'$ queries being described by term t. In the case of the intersection alternative, a document identifier was in the relevance set for a composite query for term t only if it was a member of the relevance set for all of the CF$'$ queries that were being described by term t.

Table 3.2 on the next page lists the 16 terms that occurred as the single term representation of two or more queries. The *frequency* column for a term indicates the number of queries in the CF$'$ test collection that was represented by this term, the number in the *union* column is the cardinality of the unioned sets of document identifiers for this term, and the number in the *intersection* column is the cardinality of the intersected sets of document identifiers for this term.

This paragraph contains an example that helps to explain how a composite query was constructed. Consider the term "vitamin" that appears as term #16 in Table 3.2 on the following page. It describes exactly three queries in CF$'$: Query #9, Query #10, and Query #41. The relevance set for the first query is

$$\{165, 174, 362, 370, 414, 443, 794, 992, 1040, 1115\};$$

Table 3.2: The Single Term Query Descriptions With Plural Frequencies.

line	*term*	*frequency*	*union*	*intersection*
1	acid	2	41	0
2	aeroso	2	63	27
3	aeruginosa	4	113	1
4	antibiot	2	90	30
5	class	2	71	11
6	diseas	2	104	17
7	fatti	2	52	14
8	glycoprotein	2	137	12
9	insulin	2	92	15
10	lung	2	111	33
11	pancreat	4	231	2
12	patient	4	303	4
13	polyp	2	33	2
14	saliva	2	50	5
15	sweat	5	164	0
16	vitamin	3	41	2

the set for the second query is

$$\{30, 126, 157, 170, 296, 301, 322, 370, 413, 443, 581, 676, 715, 722,$$
$$728, 758, 782, 835, 878, 941, 1115, 1215, 1218, 1234, 1239\};$$

and the one for the last query is

$$\{46, 296, 301, 322, 370, 392, 603, 941, 998, 1106, 1107, 1108, 1115, 1184, 1190\}.$$

The cardinalities of these sets are 10, 25, and 15, respectively. The union of these sets is a set with the cardinality of 41 (instead of 10+25+25=50) because some document identifiers (e.g., 301, 322, 370, 443, 941, 1115) are members of more than one of these sets. The intersection of these same 3 sets is a set with the cardinality of 2 because the only identifiers that appear in all 3 of the sets are identifiers 370 and 1115.

The restriction of the query set to only the 58 queries that corresponded to term descriptors with a frequency of exactly one was not a viable possibility for this dissertation because the number of queries with this restriction was less than 60% of the original number (i.e., too small a yield). In order to have more queries, composite queries were constructed for the 16 terms that appeared as a descriptor of more than one of the original queries. Now, the decision was: Should union (i.e., disjunctive) or intersection (i.e., conjunctive) semantics be used to construct the composite queries? Both approaches had merits, but the author did not feel that one approach was significantly superior to the other. Therefore, the author decided to create two versions of the CF$'$ test collection: CF$'_u$ (the union version) and CF$'_i$ (the intersection version).

Both versions had, as their core, the queries that corresponded to the 58 terms that had a frequency of 1. Sixteen composite queries were created for the queries with union semantics for the relevance sets. However, only 14 composite queries were constructed for the queries that have intersection semantics because lines 1 and 15 of Table 3.2 on page 70 show that the intersected relevance sets for the queries that are described by the terms "acid" and "sweat," respectively, do not have any common document identifiers. This means that the number of queries that were members of the CF'_u and CF'_i test collections were 74 (58 original single term queries + 16 composite queries) and 72 (58 original single term queries + 14 composite queries). The number of queries in the combined test collection, $\mathrm{CF}'_{\mathrm{combined}}$, was 88 (58 original single term queries + 16 composite queries from CF'_u + 14 composite queries from CF'_i).

3.2.2 Create Synthetic Datasets and Random Sets of Queries

Most of these entities were created on an as-needed basis for the work that occurs in Chapter 8 (Validation of the Formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ Measures). Chapter 10 uses synthetic data to validate many of the performance measure equations that are derived there. The specific details of these synthetic datasets and sets of queries are detailed in Chapters 8 and 10.

3.2.3 Expected Performance of the CF-related and Synthetic Test Collections

Recall-precision graphs with the standard 11-point interpolated precision, such as those used in TREC performance evaluations (Harman, 2005), can be used to express some aspects of how well a ranking algorithm performs. Later, these graphs are used to illustrate the expected performance of the Cystic Fibrosis test collection, and its three derivatives, for each of the 6 ranking methods that are used in this dissertation.

Brief review of recall-precision graphs

Before discussing the graphs, we provide a brief review of recall-precision graphs and the procedure that was used to construct them. A *recall-precision graph* for a query and its associated document collection illustrates the relationship between the recall and precision values at different recall points in a sequence of ranked documents as the recall values increase from 0.0 to 1.0, inclusive. A recall point corresponds to a position in the sequence where there is a relevant document. The recall value at a point is simply the number of relevant documents that have been encountered from the front of the sequence up to, and including, this point, divided by the total number of relevant documents that are in the collection. For example, in Table 3.3 on the following page, the third relevant document in the sequence is not encountered until position 5. Since the entire collection only has four relevant documents, the recall value at this point is 3/4=0.75.

The number of unique recall values is dependent on the number of relevant documents for a query. Therefore, it can, and often does, vary considerably, from one query to another. For example, even though there are 10 relevant documents in the collection that is associated with the data in Table 3.3 on the next page, there are only four unique recall values because the query only has four relevant documents.

Standard recall points and interpolated precision

Query-dependent variability in the number of distinct recall values is not desirable when doing performance evaluation over many queries for a collection because it complicates the evaluation process and the construction of an entity like a recall-precision graph. TREC eliminates this variability by using 11 standard recall points. These points correspond to the recall values $0.0, 0.1, \ldots, 1.0$. A question that arises concerns what the precision value is for recall value 0.0 (because precision is undefined at this point) and what the precision values are when the query has either less than, or more than, 10 relevant documents. The way that TREC evaluation software handles the latter part of the question is by interpolation. TREC evaluation software calculates the interpolated precision $p_{\text{interpolated},r'}$ at recall value r as being the maximum of the actual precision values $p_{\text{actual},r}$ that occur at recall values that are greater than, or equal to, actual recall value r (Harman, 2005; Manning et al., 2008), i.e.,

$$p_{\text{interpolated},r'} = \max_{r' \geq r} p_{\text{actual},r}.$$

A side effect of this interpolation technique is that the precision value at recall value 0.0 is now defined. Ordinarily, the precision value at this recall point does not exist because the 0.0 recall point corresponds to the situation where no documents have been examined. Therefore, the denominator of the expression that calculates precision at this point is 0. Table 3.4 on the following page enumerates the interpolated precision values for the information in Table 3.3 and Figure 3.1 on the following page shows the recall-precision graph for the information in Table 3.4 on the next page. The data for the recall-precision graph for a set of queries can be obtained in the following way: calculate the interpolated recall and precision table for each query, then use this data to compute the mean precision value at each of the 11 recall points.

Table 3.3: Actual Recall and Precision Table For A Query With Four Relevant Documents.

Rank	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Relevant?	Yes	Yes	No	No	Yes	No	Yes	No	No	No
Precision	1/1	2/2	2/3	2/4	3/5	3/6	4/7	4/8	4/9	4/10
Recall	0.25	0.50	0.50	0.50	0.75	0.75	1.00	1.00	1.00	1.00

Table 3.4: Interpolated Recall and Precision Table.

Precision	1	1	1	1	1	1	3/5	3/5	4/7	4/7	4/7
Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

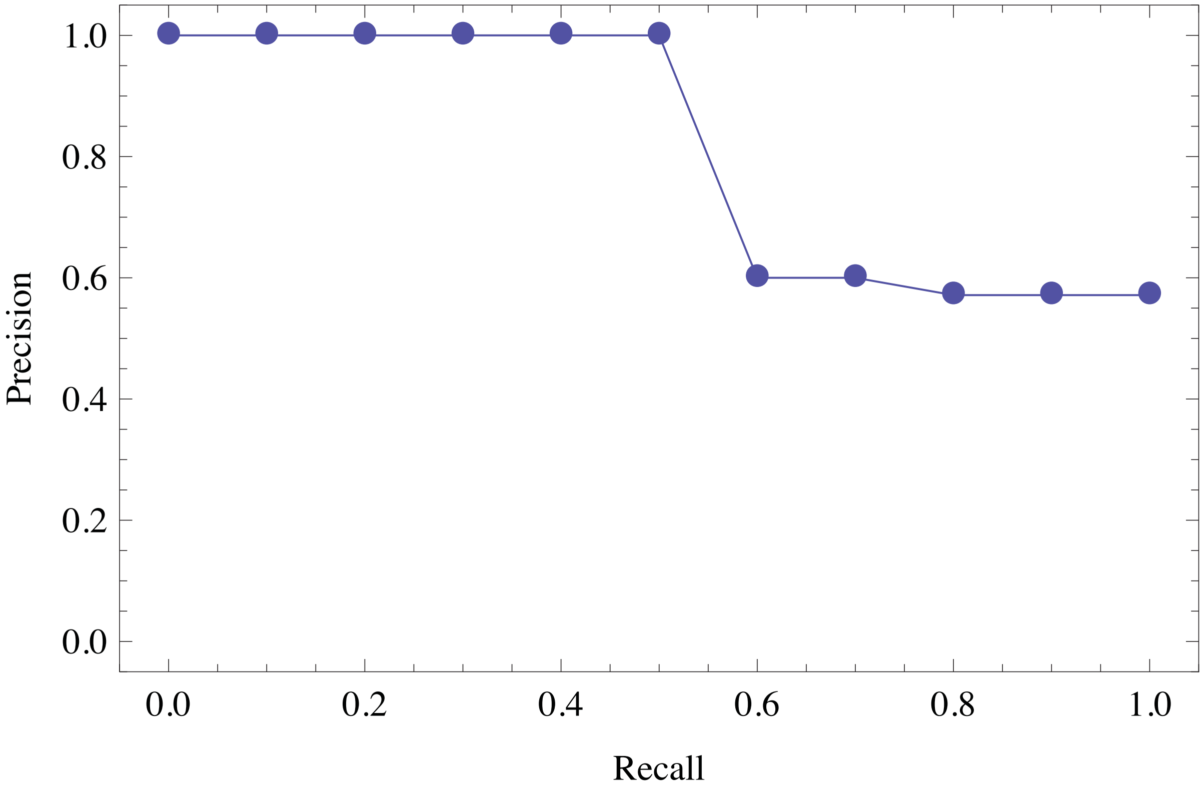


Figure 3.1: Recall-precision graph for the data in Table 3.4.

The complication of duplicate retrieval status values

Duplicate RSVs can cause complications in performance evaluations. Many performance measure evaluation algorithms do not take into account the potential presence of these duplicate values and can compute misleading results. Chapter 10 (The ASL Measure and Three Frequently-Used Performance Measures) contains detailed discussions on the impact of these duplicate RSVs and develops duplicate-sensitive (i.e., Type-T) versions of several information retrieval performance measures.

For the convenience of the reader, we repeat the definition of Type-T from page 27. The term *Type-T* is used in Chapters 2, 3, and 10 as an adjective to denote a performance measure whose calculated values are consistent with the assumption that *some* of the documents in a vector V of ranked documents *may* have tied (i.e., duplicate) RSVs. And, these chapters also use *Type-D* as an adjective to denote a performance measure whose calculated values are consistent with the assumption that *all* the documents in a vector V *must* have distinct RSVs.

More specifically, there are two main problems that must be solved before we can compute recall-precision graphs that correspond to the Cystic Fibrosis test collection, their derivatives, and the synthetic document test collection. Each of these collections have many duplicate RSVs in the rankings that are associated with each query for every one of the 6 ranking methods. Every ranking partitioned the RSVs into 2 clusters, each of which contained many duplicate values. In order to generate an actual recall and precision table for a query, test collection, and ranking method combination, we must be able to calculate the precision for the points in the ranking that each relevant document appears at in the sequence of ranked documents.

In any cluster, we assume that an arbitrary document in this cluster can equally likely occupy any of the positions in the cluster. For example, if the cluster has 5 documents, one of which is relevant, and the sub-sequence that is associated with this cluster occupies positions 4-8, inclusive, then the probability that the relevant document occupies position 4 is 1/5, the probability that it occupies position 5 is 1/5, the probability that it occupies position 6 is 1/5, the probability that it occupies position 7 is 1/5, and the probability that it occupies position 8 is 1/5. The same set of identical identical probabilities is associated with each of the other four documents in this cluster.

Table 3.5 is an example of the ranking of a document collection that has 22 documents. We use the information in this table to illustrate how to construct an "actual recall" and precision table for this information. The number of distinct RSVs in the ranking for this table is 6 (hence, the six clusters). The highest-valued RSV is 6.92 (exactly three documents have this value for their RSV) and the lowest-valued one is 0.27 (exactly seven documents have this value for their RSV).

Table 3.5: A Ranking That Has Multiple Documents With The Same RSV

Cluster	*1*	*2*	*3*	*4*	*5*	*6*
RSV of each document in the cluster	6.92	4.43	4.19	3.74	1.05	0.27
Number of Documents	3	4	2	5	1	7
Number of Relevant Documents	1	2	2	0	1	5
Position(s)	1-3	4-7	8-9	10-14	15	16-22

Construction of the Actual and Interpolated Recall and Precision Tables

The construction of the "actual precision" and recall table for the information in Table 3.5 has two primary phases. The first phase determines the *expected* actual positions of the relevant documents in each cluster. The second phase uses these expected positions to determine the respective associated precision values. In order to construct this table and, later, the corresponding interpolated recall and precision table for Table 3.5, the author has to rely on some results from Chapter 10 and on mathematics that are not introduced and discussed until then. This should not be a hindrance, though, because the main purpose of this chapter is to provide an overview of the methodology that the author intends to use to conduct the research for this dissertation, with the major details of the research being left for later chapters. Phase 1 uses results from Chapter 10 and Phase 2 uses results from Appendix C. The results that are used in the subsequent discussion are given *without proof* of the mathematics that were used to produce them. For the associated proofs and more details, the interested reader may want to consult Chapter 10 and Appendix C.

The Determination of the Expected Actual Recall Positions

We begin by calculating the expected position of a relevant document in Cluster 1. Note that this cluster has three documents, with only one of them being relevant. The three document sequence possibilities appear below. In this enumeration, the letter R denotes a relevant document whereas the letter N denotes a non-relevant document. The first letter in each row represents position 1 with the following two consecutive letters representing, respectively, positions 2 and 3.

RNN
NRN
NNR

Since it is equally likely that the relevant document can occupy any three of the positions in a sequence, the expected actual position of the relevant document in Cluster 1 is

$$(1 + 2 + 3)/3 = 6/3 = 2.$$

Cluster 2 has four documents and only two of them are relevant. In this case, there are the 6 document position sequence possibilities that are enumerated below. The first letter in each row represents position 4 with the following three consecutive letters representing, respectively, positions 5, 6, and 7.

RRNN

RNRN

RNNR

NRRN

NRNR

NNRR

Our interest here is in determining the expected actual positions of the first and second relevant documents that are encountered when the reader examines the letters in a sequence in the position order 4, 5, 6, and 7. From a visual inspection of these 6 possibilities, we see that the first relevant document occurs at position 4 three times, at position 5 two times, and at position 6 one time. This implies that the expected actual position of the *first* relevant document is

$$(4 + 4 + 4 + 5 + 5 + 6)/6 = 28/6 = 4\tfrac{2}{3}.$$

Similarly, the expected actual position of the second relevant document over these sequences is

$$(5 + 6 + 7 + 6 + 7 + 7)/6 = 38/6 = 6\tfrac{1}{3}.$$

Cluster 3 has two documents, both are relevant. The expected actual positions of their first and second relevant documents are, respectively, 8 and 9, because the only sequence possibility is the one that is right below.

RR

Cluster 4 has five documents but none of them are relevant. Therefore, we need not be concerned with determining the expected actual position of a relevant document because none exist. Cluster 5 has one document and it is also relevant. The sole sequence possibility that is associated with it appears immediately below.

$$\boxed{\text{R}}$$

The expected actual position of this document is 15. Cluster 6 has seven documents, only five of them are relevant. There are $\binom{7}{5} = 21$ distinct sequence possibilities for this cluster and five expected actual positions of relevant documents. Instead of enumerating these 21 possibilities, and determining these 5 positions empirically, we analytically determine them by the results of Lemma C.0.1 on page 535. We state below, without proof, this lemma.

Lemma C.0.1. *Suppose $1 \leq i \leq r \leq n$ and $i, r, n, l \in \mathbb{N}$. Let $[l, l+n-1]$ represent positions $l, l+1, \ldots, l+n-1$ in an equivalence class of n documents with exactly r relevant documents. Assuming that a relevant document has the same probability of occupying any one of these n positions as it does of occupying any one of the other $n-1$ positions, the expected mean position for the ith relevant document from the beginning of the interval is*

$$i - 1 + l + i(n-r)/(r+1).$$

Before using this lemma to determine the five expected positions that are associated with the relevant documents of Cluster 6, let us use it to calculate the expected positions for Clusters 1, 2, 3, and 5. We demonstrate that these sets of analytically-determined positions are equal to those that we just obtained empirically by exhaustive enumeration. This should give us confidence that we can use the analytic method to generate the correct results for Cluster 6.

For Cluster 1, we have $i = l = 1$, $n = 3$, and $r = 1$. This means that

$$\begin{aligned} i - 1 + l + i(n-r)/(r+1) &= 1 - 1 + 1 + 1(3-1)/(1+1) \\ &= 1 + 2/2 \\ &= 2. \end{aligned}$$

This position is identical to the position that was calculated earlier by empirical means.

For Cluster 2, we have i varying from 1 to 2, inclusive, with $l = n = 4$, and $r = 2$. This means that, when $i = 1$,

$$\begin{aligned} i - 1 + l + i(n-r)/(r+1) &= 1 - 1 + 4 + 1(4-2)/(2+1) \\ &= 4 + 2/3 \\ &= 4\tfrac{2}{3} \end{aligned}$$

and, when $i = 2$,

$$\begin{aligned} i - 1 + l + i(n-r)/(r+1) &= 2 - 1 + 4 + 2(4-2)/(2+1) \\ &= 5 + 4/3 \\ &= 6\tfrac{1}{3}. \end{aligned}$$

These two positions are identical to the positions that were calculated earlier by empirical means.

For Cluster 3, we have i varying from 1 to 2, inclusive, with $l = 8$ and $n = r = 2$. This means that, when $i = 1$,

$$\begin{aligned} i - 1 + l + i(n-r)/(r+1) &= 1 - 1 + 8 + 1(2-2)/(2+1) \\ &= 8 + 0 \\ &= 8 \end{aligned}$$

and, when $i = 2$,

$$\begin{aligned} i - 1 + l + i(n-r)/(r+1) &= 2 - 1 + 8 + 2(2-2)/(2+1) \\ &= 9 + 0 \end{aligned}$$

$$= 9.$$

These two positions are identical to the positions that were calculated earlier by empirical means.

For Cluster 5, we have $i = 1$, $l = 15$, and $n = r = 1$. This means that

$$\begin{aligned} i - 1 + l + i(n - r)/(r + 1) &= 1 - 1 + 15 + 1(1 - 1)/(1 + 1) \\ &= 15 + 0 \\ &= 15. \end{aligned}$$

This position is identical to the position that was calculated earlier by empirical means.

Finally, for Cluster 6, we can follow a similar procedure to those above to obtain the set $\{16\frac{1}{3}, 17\frac{2}{3}, 19, 20\frac{1}{3}, 21\frac{2}{3}\}$ of expected actual positions. These positions constitute the last set of positions that we needed to determine for our example. Next, we must obtain the precision value that is associated with each of these positions.

The Determination of the Interpolated Precision Values for An Expected Actual Recall Position

Generally, the expected actual recall positions are not whole numbers. For example, four of the expected actual positions (i.e., $16\frac{1}{3}, 17\frac{2}{3}, 20\frac{1}{3}, 21\frac{2}{3}$) for Cluster 6 are not whole numbers. When the expected position is a whole number, we use the Type-T version of the precision equation, which is located on page 460, to calculate the precision at that position in the ranking.

However, when the position (e.g., $16\frac{1}{3}$) has a fractional part, the precision value at this expected position must be interpolated. The interpolation process works by first determining the closest whole numbers l and g that are, respectively, less than this position and greater than this position. For example, if the position was $16\frac{1}{3}$, these numbers would be, respectively, 16 and 17, and, if the position was $21\frac{2}{3}$, these numbers would be, respectively, 21 and 22. The next step in this process is to use the Type-T version of the precision measure to calculate the precision values at positions l and g. Let the corresponding precision values be denoted by p_l and p_g. If we use the variable e to denote the expected position, then we can use linear interpolation to approximate the precision value at e. The approximated value is very accurate given that the l and g are always adjacent points.

Harris and Stöcker (1998) states that the value for $f(x)$, where $x_1 < x < x_2$, and $f(x_1)$ and $f(x_2)$ are known values, can be determined by this equation:

$$f(x) = \frac{(x_2 - x)f(x_1) + (x - x_1)f(x_2)}{x_2 - x_1}.$$

Based on this equation, we obtain the following equation for computing the interpolated precision value when the expected position is strictly between l and g (i.e., the open interval (l, g)):

$$\begin{aligned} p_i &= \frac{(g-e)p_l + (e-l)p_g}{g-l} \\ &= \frac{(g-e)p_l + (e-l)p_g}{1} \\ &= (g-e)p_l + (e-l)p_g. \end{aligned}$$

In general,

$$p_i = \begin{cases} p_l, & \text{if } \lfloor e \rfloor = \lceil e \rceil; \\ (g-e)p_l + (e-l)p_g, & \text{otherwise;} \end{cases}$$

where $\lfloor e \rfloor$ denotes the greatest integer that is less than or equal to e and $\lceil e \rceil$ denotes the least integer that is greater than or equal to e.

Culmination of the Example

From the earlier discussions, we know that the set of expected actual positions is

$$\left\{2, 4\tfrac{2}{3}, 6\tfrac{1}{3}, 8, 9, 15, 16\tfrac{1}{3}, 17\tfrac{2}{3}, 19, 20\tfrac{1}{3}, 21\tfrac{2}{3}\right\}.$$

We can use the Type-T precision method that is developed in Chapter 10 to calculate the precision value for each integer in this set. This method, in conjunction with our method to interpolate precision when the expected actual position in not an integer, results in the information in Table 3.6. The information for the associated interpolated recall and precision tables appears in Table 3.7.

Table 3.6: Expected Actual Recall Position (EARP) Table.

Precision	0.333	0.392	0.421	0.5	0.556	0.4	0.426	0.447	0.466	0.482	0.497
Recall	1/11	2/11	3/11	4/11	5/11	6/11	7/11	8/11	9/11	10/11	1
EARP	2	$4\frac{2}{3}$	$6\frac{1}{3}$	8	9	15	$16\frac{1}{3}$	$17\frac{2}{3}$	19	$20\frac{1}{3}$	$21\frac{2}{3}$

Table 3.7: Interpolated Recall and Precision Table.

Precision	0.556	0.556	0.556	0.556	0.556	0.497	0.497	0.497	0.497	0.497	0.497
Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

Performance graphs

The performance graphs for the best-case, coordination-level, decision-theoretic, and inverse document frequency ranking methods, for the CF test data, were identical due to several primary factors: each of the respective rankings contained large numbers of duplicate retrieval status values; each ranking had, at most, two distinct RSVs; the influence of binary relevance; and term weights were based upon a term being either

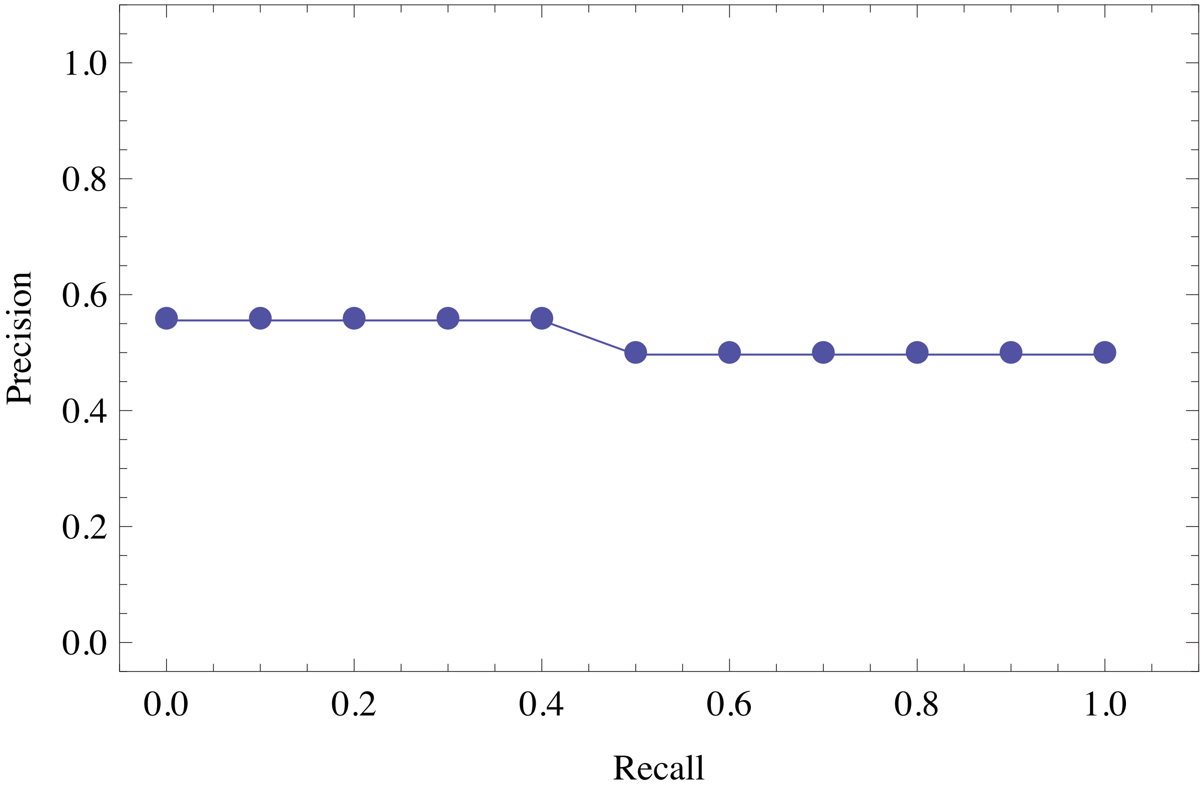


Figure 3.2: Recall-precision graph for the data in Table 3.7 on the previous page.

present or absent (i.e., multiple occurrences of a term in a document have the same weight as a solitary occurrence of the term). From a ranking perspective, even though different RSV weights were associated with each of these ranking methods, the overall rankings had identical recall-precision performance characteristics. For a particular ranking, say, the best-case ranking method, the largest performance differences occurred for recall values that were less than or equal to 0.5. The precision values were effectively identical for the recall values that were greater than 0.5.

The shapes of the curves in the performance graph for random-case ranking were very similar to those for the four ranking methods that were discussed in the immediately prior paragraph. The main differences for random-case ranking were that there was slightly more variability in the precision values at each recall-precision point and that, overall, the precision value at each point appeared to be about 0.15 lower than the corresponding values for the curves in the first two rows of graphs in Figure 3.3 on page 88. The curves in the graph for the worst-case ranking method differed significantly from those of the other 5 ranking methods. Notice that the "curves" were actually lines, that the precision values different for each curve but is a constant with respect to a particular curve. Furthermore, these precision values were near zero and differed by approximately an order of magnitude from the precision values at the corresponding recall-precision points in the first 2 rows of graphs. The differences were not as great when the curves from this worst-case ranking were compared to their random-case ranking counterparts.

To summarize, the information in the graphs for the first two rows of ranking methods indicated that the test collections for these methods, on a performance basis, should be ranked from best to worst in this order: CF'_u, $\mathrm{CF}'_{\mathrm{combined}}$, CF'_i, and CF'. The information in Figure 3.3 on page 88 also indicated that the best to worst performance ordering should be $\mathrm{CF}'_{\mathrm{combined}}$, CF'_u, CF', and CF'_i. For the worst-case ranking method, the indicated ordering is CF'_u, $\mathrm{CF}'_{\mathrm{combined}}$, CF', and CF'_i.

Conceptually, the only difference between the 6 performance graphs that are illustrated in Figure 3.3 on the following page, and those in Figure 3.4 on page 89, is that Figure 3.4 also includes performance information about the synthetic test collection. Notice that the synthetic test collection curves for the best-case, coordination-level matching, decision-theoretic, inverse document frequency, and random-case ranking methods indicated that the precision values had a small gradual decrease as the recall values increased and that the ending precision value (at recall value 1.0) for such a curve did not differ that much, percentage-wise, from the initial value (at recall value 0.0). The curve for the worst-case ranking method was linear. This was due to the combination of the actual data values and the algorithm that TREC used to interpolate precision values. Overall, regardless of ranking method, and at each of the 11 standard recall points, one can expect significantly better retrieval performance for the queries in the synthetic document test collection than for those queries in any of the other four test collections.

3.3 Quality of Ranking Calculations for the Coordination Level Matching, Inverse Document Frequency, and Decision-Theoretic Ranking Methods

Losee (1998) states that the equation for $\mathcal{Q}$ (the degree of optimality) for both the basic version of inverse document frequency (IDF) algorithm and the coordination level matching (CLM) algorithm is the same; that is,

$$\mathcal{Q}_{\mathrm{IDF}} = \mathcal{Q}_{\mathrm{CLM}} = \Pr(p > t), \tag{3.3.1}$$

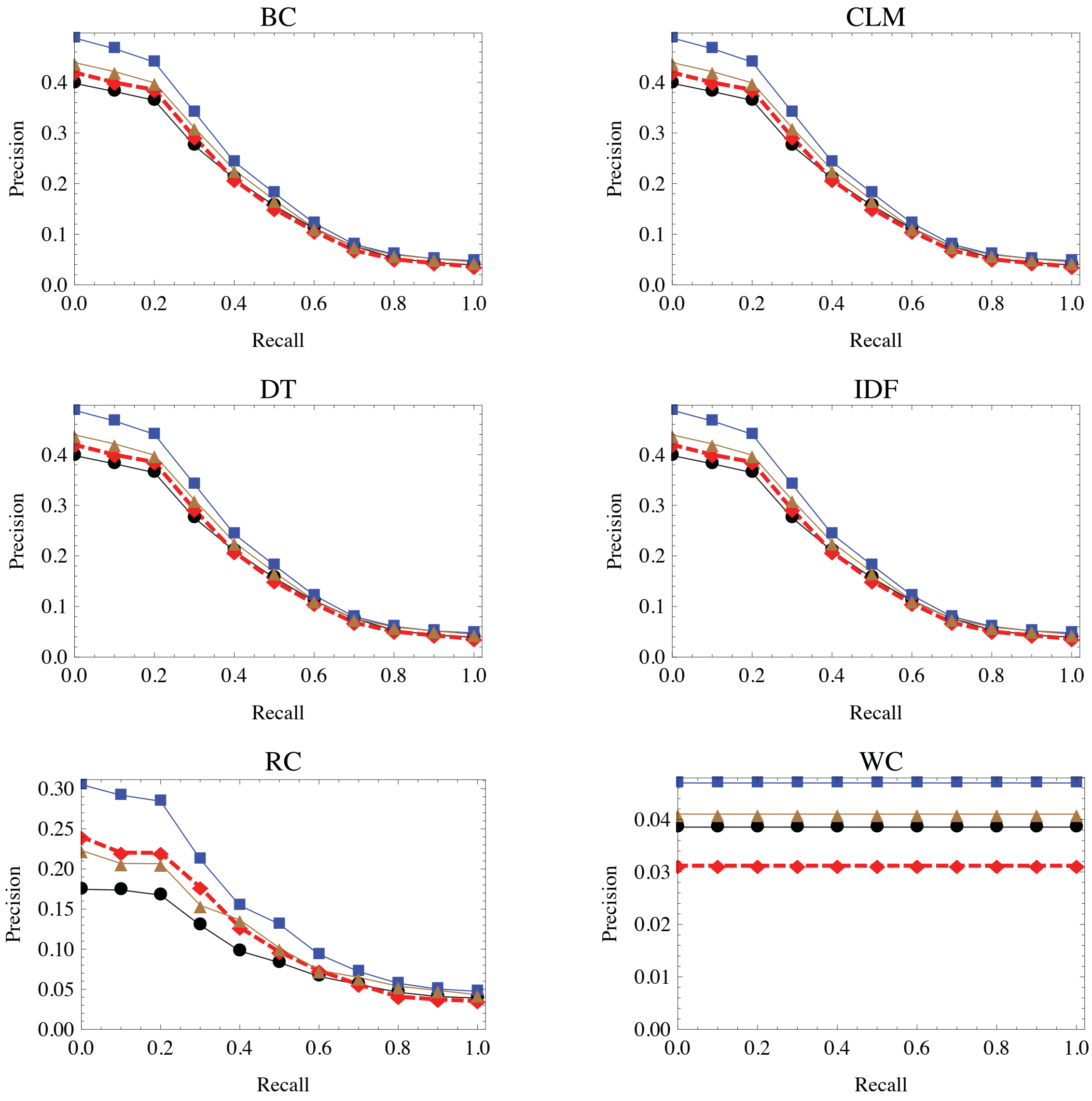


Figure 3.3: Recall-precision graphs for the four derivatives of the Cystic Fibrosis test collection. Each derivative collection has 1239 documents. The number of queries that are in the CF$'$, CF$'_u$ and CF$'_i$ test collections are, respectively, 100, 74, and 72. The number of queries that are in the combined test collection, CF$'_{\text{combined}}$, is 88. In the plots, the recall-precision curves CF$'$, CF$'_u$ and CF$'_i$, and CF$'_{\text{combined}}$ collections, respectively, are represented by a black curve with the recall-precision points represented by circles, a blue curve with the recall-precision points represented by squares, a red dashed curve with the recall-precision points represented by circles, and a brown curve with the recall-precision points represented by triangles that point upward. Note that the precision axes are the same for the first two rows of this figure but are different for the last two rows.

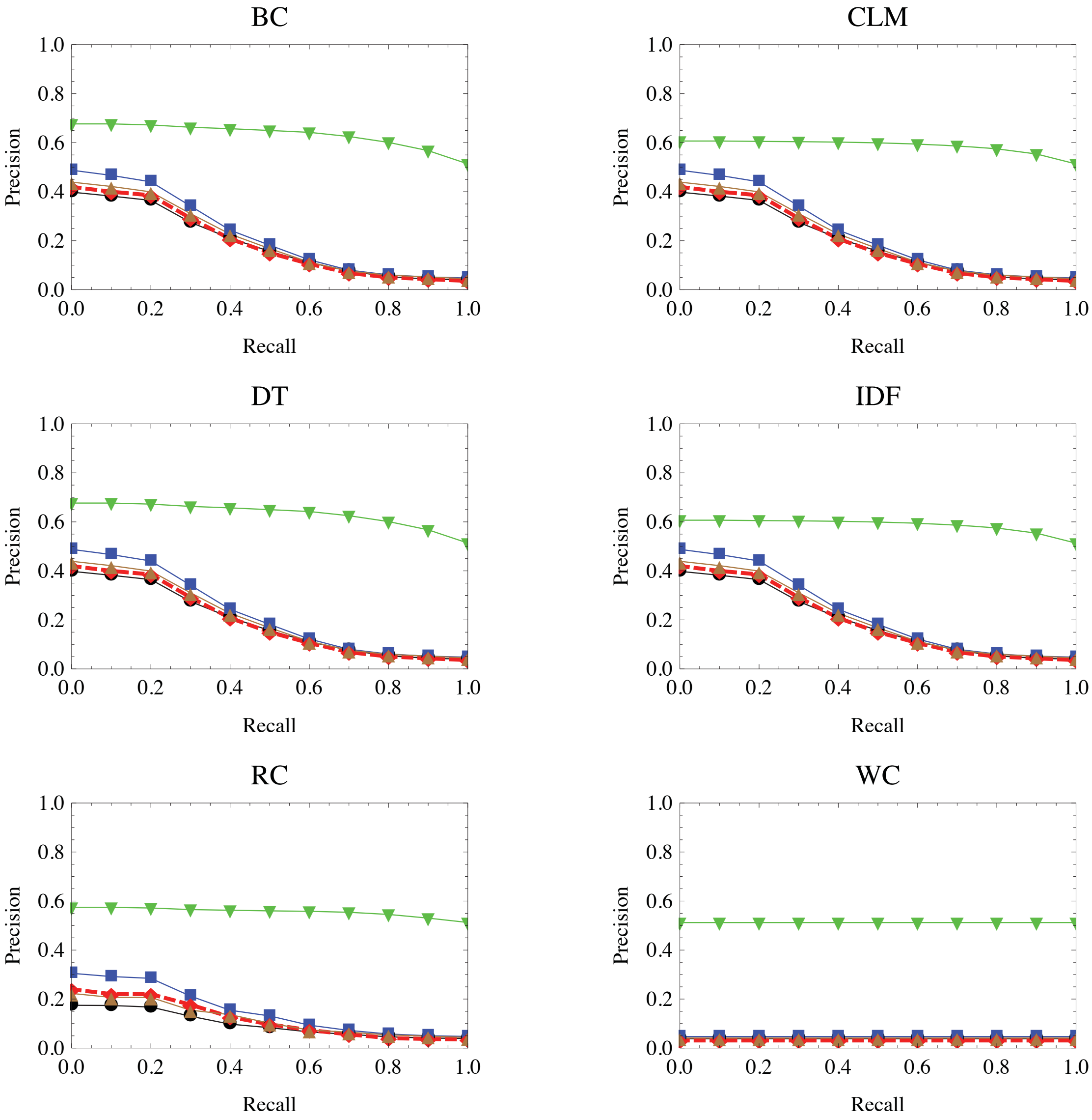


Figure 3.4: Recall-precision graphs for the four derivatives of the Cystic Fibrosis test collection and a synthetic test collection. Each collection has 1239 documents. The number of queries that are in the CF$'$, CF$'_u$ and CF$'_i$ test collections are, respectively, 100, 74, and 72. The number of queries that are in the combined test collection, CF$'_{\text{combined}}$, is 88. The number of queries that are in the synthetic test collection is 100. In the plots, the recall-precision curves CF$'$, CF$'_u$ and CF$'_i$, and CF$'_{\text{combined}}$ collections, respectively, are represented by a black curve with the recall-precision points represented by circles, a blue curve with the recall-precision points represented by squares, a red dashed curve with the recall-precision points represented by circles, and a brown curve with the recall-precision points represented by triangles that point upward. The recall-precision curve for the synthetic test collection is represented by the green curve with triangles that point upward. Over all of the ranking methods, each precision component of the recall-precision points for the curves that are associated with the synthetic test collection has a precision value that is greater than 0.5.

where $p = \Pr(d|rel)$ is the probability of a particular feature with frequency 1 occurring in a relevant document and $t = \Pr(d)$ is the probability of that feature with frequency 1 unconditionally occurring in a document. In this dissertation the word "feature" is synonymous with the phrase "query term." Therefore, p can be interpreted as the probability that a relevant document contains the query term and t can be interpreted as the probability that any document contains the query term.

This dissertation, however, used a slightly different equation for the $\mathcal{Q}_{\text{IDF}}$ measure. The very minor difference between Equation 3.3.1 on page 87 and Equation 3.3.2 for this measure was the handling of a boundary condition when $t = 1$. The alternate formulation was

$$\begin{aligned} \mathcal{Q}_{\text{IDF}} &= \Pr(p > t, 0 < t < 1) + \Pr(p \leq t, t = 1) \\ &= \Pr(p > t) + \Pr(p \leq t, t = 1). \end{aligned} \tag{3.3.2}$$

Its derivation is discussed in Appendix D.

In this research, a feature frequency of 1 or 0 corresponded, respectively, to the presence or absence of a single word term in a query or document.

Losee (1998) also states that the equation for $\mathcal{Q}$, for a decision-theoretic (DT) ranking method based on binary independent features, is

$$\mathcal{Q}_{\text{DT}} = \Pr(p > \max(t, q)) + \Pr(p \leq \min(t, q)), \tag{3.3.3}$$

where $q = \Pr(d|\overline{rel})$ is the probability of a particular feature with frequency 1 occurring in a non-relevant document. The expression $\max(t, q)$ denotes the maximum of t and q and $\min(t, q)$ denotes the minimum of those values. In addition to the notation introduced in Chapter 2, page 21, Section 2.2.5 (The Query-Document Model), let $s_i = n_i - r_i$, denote the number of non-relevant documents with feature frequency $i \in \{0, 1\}$.

Weak compositions were used to help construct models to study some performance aspects of versions of the coordination level matching (CLM), inverse document frequency (IDF), and decision-theoretic (DT) ranking algorithms that appear in Losee (1998). Three of the primary interests in this research were how to use Equation 3.3.1 on page 87, Equation 3.3.2 on the preceding page, and Equation 3.3.3 on the previous page to calculate the quality of ranking measures for the coordination level matching, inverse document frequency, and the decision-theoretic ranking methods, respectively. Since p can be expressed as

$$\frac{r_1}{r_0 + r_1}, \tag{3.3.4}$$

t can be expressed as

$$\frac{r_1 + s_1}{r_0 + r_1 + s_0 + s_1}, \tag{3.3.5}$$

and q can be expressed as

$$\frac{s_1}{s_0 + s_1}, \tag{3.3.6}$$

this means that part of the answer to Equation 3.3.1 on page 87, Equation 3.3.2 on the preceding page, and Equation 3.3.3 on the previous page can be modeled by a set containing all the weak compositions of size 4 such that

$$N = r_0 + r_1 + s_0 + s_1;$$

this set is denoted by apwc_{4N} (the set of *all possible weak compositions of size 4 for* N). With respect to Equation 2.2.2 on page 26, the number of weak compositions in apwc_{4N}

is

$$\tilde{C}_4(N) = \binom{N+3}{3}.$$

The combinatoric-based quality of ranking formulas that were developed for the CLM ranking model (shown by Chapter 5) and the IDF ranking model (shown by Section 6.1) used the parameters p and t; the analogous model for DT ranking (shown by Section 6.3) used the parameters p, t, and q. In the models associated with Equations 3.3.1 to 3.3.3 on pages 87–90, the values of p, q, and t were not defined for all of the outcomes in the sample space of weak 4-compositions for a collection of $N \geq 0$ documents. Chapter 4 contains a discussion of several techniques for handling singularities.

A weak 4-composition is represented by a 4-tuple of this form: (r_1, s_0, r_0, s_1). Using the formulas given for p and t a few paragraphs back, we can express the relation (shown by Inequality 3.3.7) that must hold between the r_1, s_0, r_0, and s_1 values in any weak composition for $p > t$ to be true for it. This relationship is not required to hold for every weak composition in set $\mathrm{apwc}_{\mathrm{4N}}$. It must hold, though, for every weak composition that will contribute a count of 1 to the count of the total number of weak compositions that meet the criterion $p > t$. The main idea here is to compute $\Pr(p > t)$ by determining the number of weak compositions in $\mathrm{apwc}_{\mathrm{4N}}$, then dividing that number by the cardinality of that set. An example illustrating how to do that appears later in this section.

In Equation 3.3.1 on page 87, if $p > t$ is true, then

$$\frac{r_1}{r_0 + r_1} > \frac{r_1 + s_1}{r_0 + r_1 + s_0 + s_1} \tag{3.3.7}$$

must also be true because that relationship can be obtained by simply substituting

$$\frac{r_1}{r_0 + r_1}$$

for p and by substituting

$$\frac{r_1 + s_1}{r_0 + r_1 + s_0 + s_1}$$

for t in the expression $p > t$.

After cross-multiplying the corresponding numerator/denominator pairs in Inequality 3.3.7 on the preceding page, we obtain

$$r_1(r_0 + r_1 + s_0 + s_1) > (r_0 + r_1)(r_1 + s_1). \tag{3.3.8}$$

After expansion of the expressions on both sides of the greater-than operator in Inequality 3.3.8 , we have

$$r_1r_0 + r_1^2 + r_1s_0 + r_1s_1 > r_0r_1 + r_0s_1 + r_1^2 + r_1s_1. \tag{3.3.9}$$

Note that, in Inequality 3.3.9, the first, second, and fourth terms on the left-hand side of the greater-than operator are equal to the first, third, and fourth terms, respectively, on the right-hand side of that operator. If we cancel the equivalent terms, we have

$$\cancel{r_1r_0} + \cancel{r_1^2} + r_1s_0 + \cancel{r_1s_1} \quad > \quad \cancel{r_0r_1} + r_0s_1 + \cancel{r_1^2} + \cancel{r_1s_1}\,. \tag{3.3.10}$$

After setting the canceled terms to 0, and then simplifying, we obtain

$$r_1s_0 > r_0s_1.$$

This relationship can be used to help calculate the total number of events (i.e., the total number of weak compositions (i.e., document collections) where $p > t$) in the sample space that we are concerned with. The number that qualify can be represented by the number of events where the following relationships hold:

$$r_1 s_0 > r_0 s_1 \tag{3.3.11}$$

$$r_0 + r_1 + s_0 + s_1 = N \tag{3.3.12}$$

$$0 \leq r_0, r_1, s_0, s_1 \leq N \tag{3.3.13}$$

$$r_0, r_1, s_0, s_1, N \in \mathbb{N}. \tag{3.3.14}$$

This set of constraints can be studied with combinatorial structures and identities. Basically, it can be modeled with weak compositions of size 4 subject to the constraints just given above. The symbol $\mathbb{N}$ denotes the set of natural numbers.

In this dissertation, the performance of the coordination level matching (CLM), inverse document frequency (IDF), and decision-theoretic (DT) ranking measures for a corpus with N documents, was investigated by using weak compositions of size 4, subject to certain constraints. For the three measures that are currently being discussing, several steps were taken in order to calculate $\mathcal{Q}$. First, the number of qualifying weak compositions (denoted by numQualifiers) that satisfied Constraints 3.3.11 to 3.3.14 on the current page had to be determined. Second, the value of $\tilde{C}_4(N)$ (denoted by numPossible), the number of all possible weak compositions of size 4 for N, had to be determined. Third, the value of numQualifiers/numPossible (denoted by $\mathcal{Q}$) had to be determined.

Of the three steps that were just specified above, the process of determining a formula, or algorithm, to calculate the number of qualifying weak compositions was where, by far, the most effort was expended. From preliminary research, it was discovered that the problem of determining a formula for this value could be broken into several subproblems, of which only some had *closed form* (the next paragraph contains a definition) solutions. There did not appear to be a general formula, of simple or moderate complexity, for calculating the number of weak compositions of size 4 for an arbitrary natural number subject to Constraints 3.3.11 to 3.3.14 on the preceding page.

A *closed form* (short for *closed formula*) solution to an equation is one where the number of steps to evaluate the formula is independent of the values of its parameters. A simple motivating example is a one-parameter function f (e.g., the summation-of-positive-integers function) defined as

$$f(n) = \sum_{i=1}^{n} i, \tag{3.3.15}$$

where n is the number of integers that are being summed. These integers are positive integers that range in value from 1 to n, inclusive. If n has the value 0, then

$$\begin{aligned} f(0) &= \sum_{i=1}^{0} i \\ &= 0, \end{aligned}$$

because the starting point (i.e., 1) of the index i is greater than the summation limit (i.e., 0). A non-trivial example is the calculation of the sum of the values of the first 15 positive integers; that is,

$$\begin{aligned} f(15) &= \sum_{i=1}^{15} i \\ &= 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 \\ &= 120. \end{aligned}$$

The evaluation of this sum involves 14 additions. Suppose we want to perform a calculation that finds the sum of the first $n \geq 1$ positive integers and that binary addition is the only operation that we are allowed to use to sum the numbers. The calculation of this sum involves $n - 1$ additions. In general, if the summation limit n is a positive integer, the number of additions that is necessary to calculate the sum is $n - 1$, a value that is one less than the number of values that we wish to sum.

A much better way to go about this is to make use of a well-known summation identity. A function f_2 that uses this identity can be defined as

$$f_2(n) = n(n+1)/2.$$

If we use f_2 to calculate the same sum as above, we obtain

$$\begin{aligned} f_2(15) &= 15(15+1)/2 \\ &= 15 \cdot 8 \\ &= 120, \end{aligned}$$

the same value that was determined earlier by f, but now in a much more efficient manner.

Notice that the evaluation of f_2 only requires three arithmetic operations: an addition, a multiplication, and a division. So, independent of the value of any positive integer n, the sum of the first n positive integers for that value of n can always be determined by a single application of an addition, a multiplication, and a division. This alternate summation formula is the more desirable one to use because the number of operations to evaluate it is fixed at 3, whereas the number of operations to evaluate Equation 3.3.15 monotonically increases as the value of n increases.

During the many derivations that occur in the later chapters of this dissertation, the emphasis is always on obtaining the final equation, or sets of equations, in closed form. This is not always possible, but it is definitely a goal for these derivations. It was demonstrated earlier that one benefit of a closed form equation was a hard bound, independent of the values of the parameter(s) of a function, on the number of steps that it took to calculate its value. The other major benefit of a closed form expression is that it is generally more analytically-tractable than a non-closed form version.

As an aside, this summation defined by both functions f and f_2 can be expressed from a combinatorial perspective as $\binom{n+1}{2}$. The identity for this fact is used very often in many of the formula derivations that appear in later chapters. We revisit this identity, along with some other useful identities that appear in many of the derivations, in those chapters.

3.4 An Example of How to Estimate $\mathcal{Q}$ for the CLM Ranking Method

This example shows how to estimate $\mathcal{Q}$ for a small document collection using combinatorics and counting. It discusses how estimated values of $\mathcal{Q}$ can provide guidance as to which ranking method might be the preferred one in a particular situation. It prescribes a way of determining whether the $\mathcal{Q}$ predicted by analytical means is in close agreement with that determined by random sampling and empirical means. This example concludes by showing that the use of an exhaustive (e.g., brute force) technique, such as combinatorial enumeration, to help estimate $\mathcal{Q}$, is rather limited because the maximum number of weak compositions that we must examine roughly increases by a factor of 8 every time the number of documents in the collection doubles. Basically, this means that as the number of documents that we are modeling increases, it eventually leads to a problem called *combinatorial explosion* (Reingold et al., 1977). A combinatorial explosion, in mathematics, describes the effect of functions that have fantastic growth rates as the size of their input(s) increase. The well-known factorial function that is often encountered in probability and statistics courses is one such example.

In this example, we show how to estimate $\mathcal{Q}$ for the CLM ranking method when the document collection has 4 documents and thus a cardinality of 4. Combinatorial generation creates a set of $\tilde{C}_4(4) = \binom{7}{3} = 35$ weak compositions of size 4 (described in Table 3.8 on the next page). Combinatorial enumeration determines that there are 35 elements; it does not explicitly generate the set. These elements represent all the possible ways that 4 documents can be distributed among the 4 cells in the contingency table of Figure 2.2 on page 25. The model used to estimate $\mathcal{Q}$ assumes that each element of the set is equally likely.

Combinatorial enumeration, combined with Constraint 3.3.11 on page 94, determines that there are 9 weak compositions that qualify (the qualifying weak compositions in Table 3.8 on the following page have a 'yes' in the last column of the row corresponding to them). This is the value that is assigned to numQualifiers. The value assigned to numPossible is 35. Hence, the estimated value for $\mathcal{Q}$ is $\frac{9}{35} = 0.257143$. Since the $\mathcal{Q}$ value for an optimal ranking is defined to be 1 (Losee, 1998), this means that the degree of overlap between the rankings produced by an IDF ranking algorithm and the optimal ranking algorithm is approximately 25.7%. Another way of interpreting that number is that, on average, an IDF ranking algorithm performs only about 25.7% as well as the optimal ranking algorithm with respect to a 4-document collection.

$\mathcal{Q}$ values can be used to help decide which of several ranking algorithms would be the best one to use in certain situations. To make this discussion more meaningful, let us assume that we have a 10,000 document collection, that the $\mathcal{Q}$ value for CLM ranking is 0.48 for this collection size, that we have 2 other ranking methods, namely, Ranking Method A and Ranking Method B, and that their respective $\mathcal{Q}$ values are 0.5 and 0.75 for this collection size. In this case, Ranking Method B is the best choice because there is much more overlap between its rankings (as contrasted with those of CLM ranking and Ranking Method A) and those of the optimal ranking method.

How do we know that a $\mathcal{Q}$ that was calculated by, say, analytical means is correct, or right, for a specified collection size? One way of determining that relies on empirical

Table 3.8: Sample Space for a 4 Document Collection.

	weak composition	r_1	s_0	r_0	s_1	$r_1 s_0 > r_0 s_1$?
1	(0, 0, 0, 4)	0	0	0	4	*no*
2	(0, 0, 1, 3)	0	0	1	3	*no*
3	(0, 0, 2, 2)	0	0	2	2	*no*
4	(0, 0, 3, 1)	0	0	3	1	*no*
5	(0, 0, 4, 0)	0	0	4	0	*no*
6	(0, 1, 0, 3)	0	1	0	3	*no*
7	(0, 1, 1, 2)	0	1	1	2	*no*
8	(0, 1, 2, 1)	0	1	2	1	*no*
9	(0, 1, 3, 0)	0	1	3	0	*no*
10	(0, 2, 0, 2)	0	2	0	2	*no*
11	(0, 2, 1, 1)	0	2	1	1	*no*
12	(0, 2, 2, 0)	0	2	2	0	*no*
13	(0, 3, 0, 1)	0	3	0	1	*no*
14	(0, 3, 1, 0)	0	3	1	0	*no*
15	(0, 4, 0, 0)	0	4	0	0	*no*
16	(1, 0, 0, 3)	1	0	0	3	*no*
17	(1, 0, 1, 2)	1	0	1	2	*no*
18	(1, 0, 2, 1)	1	0	2	1	*no*
19	(1, 0, 3, 0)	1	0	3	0	*no*
20	(1, 1, 0, 2)	1	1	0	2	*yes*
21	(1, 1, 1, 1)	1	1	1	1	*no*
22	(1, 1, 2, 0)	1	1	2	0	*yes*
23	(1, 2, 0, 1)	1	2	0	1	*yes*
24	(1, 2, 1, 0)	1	2	1	0	*yes*
25	(1, 3, 0, 0)	1	3	0	0	*yes*
26	(2, 0, 0, 2)	2	0	0	2	*no*
27	(2, 0, 1, 1)	2	0	1	1	*no*
28	(2, 0, 2, 0)	2	0	2	0	*no*
29	(2, 1, 0, 1)	2	1	0	1	*yes*
30	(2, 1, 1, 0)	2	1	1	0	*yes*
31	(2, 2, 0, 0)	2	2	0	0	*yes*
32	(3, 0, 0, 1)	3	0	0	1	*no*
33	(3, 0, 1, 0)	3	0	1	0	*no*
34	(3, 1, 0, 0)	3	1	0	0	*yes*
35	(4, 0, 0, 0)	4	0	0	0	*no*

techniques and is briefly described in this paragraph. First, randomly choose M queries and N documents from a test collection. Assuming, of course, that the test collection has at least that many documents and queries. The information in the test collection can be regarded as historical data; we can use it to estimate $\mathcal{Q}$. Apply stemming and stopword removal to each query and to each document. Represent each relevance judgment as a binary relevance judgment, if necessary. Since the queries are likely to be multiple term queries, they need to be transformed to single term queries. A procedure for effecting the above transformations is described in Section 3.2.1. Next, compute p (the probability that the query term appears in a relevant document) and t (the proportion of documents that the query term appears in). If $p > t$ is true for a query, then it contributes a value of 1 to a tally of how many queries this condition is true for. The $\mathcal{Q}$ estimate from historical data is that tally divided by the number of queries. For example, if M=500 and for 307 queries $p > t$ was true, then the estimated $\mathcal{Q}$ value is $307/500 = 0.614$. Finally, one could use the Kolmogorov-Smirnov goodness-of-fit test (Conover, 1999) to determine if the distribution functions associated with these two $\mathcal{Q}$ values, one calculated by combinatoric techniques, the other calculated from historical data, are similar. A sketch of how to do this appears in the paragraph immediately below.

Use the formulas associated with the combinatoric techniques to generate the $\mathcal{Q}$ values for a 1-document collection, a 2-document collection, and so on, continuing up to, and including, an N-document collection. At the end of this process, we have N values – one for each possible size (i.e., the number of documents) of the document collection. For example, if N=100 documents, then we have a $\mathcal{Q}$ value for a 1-document collection, a possibly different $\mathcal{Q}$ value for a 2-document collection, and so on, up to, and including, a 100-document collection. This collection of 100 $\mathcal{Q}$ values is used to construct what Conover (1999) refers to as "[a] hypothesized distribution function" for $\mathcal{Q}$. The method that was just described is a method to obtain *one* of the inputs for the

Kolmogorov-Smirnov goodness-of-fit test. The other required inputs for this test are the values that correspond to what Conover calls the "empirical distribution function." These values can be obtained as follows: Assume that the test collection has N documents, M queries, and associated binary relevance judgments. For each collection size cs, from 1 to N, inclusive: randomly choose cs documents, without replacement, and randomly choose $nq \in \{1, 2, ..., M\}$ queries, without replacement, from the test collection. Use the procedure described in the previous paragraph to estimate $\mathcal{Q}$ for each of the N possible collection sizes. If N=100, we now have 100 data points, calculated from "historical data." These data points correspond to the empirical distribution function. Finally, we can use techniques described in a general nonparametric statistics text (e.g., Conover (1999)) or a simulation modeling and analysis one (e.g., Law (2006)) to determine how good the fit is.

Note that the references to combinatorial generation and enumeration above are for illustrative purposes. This research developed analytic formulas for calculating the number of qualifying compositions for particular scenarios. It envisioned using combinatorial generation and enumeration to assist in validating whatever formulas it derived. In general, however, the use of a brute force combinatorial technique such as combinatorial generation is only feasible when modeling moderate size (e.g., several hundred) document collections. This is because, for a fixed number of parts, the number of weak compositions generated grows very rapidly in terms of the number of documents in a collection. This quickly leads to the software experiencing running-out-of-memory and processor-time issues.

Here is an illustration of the growth rate of the number of weak compositions for a fixed size k. Without loss of generality, let us assume that $k = 4$. This means that

$\tilde{C}_4(N)$, the number of weak compositions of size 4 for a collection of N documents, is

$$\begin{aligned}\tilde{C}_4(N) &= \binom{N+4-1}{4-1} \\ &= \binom{N+3}{3} \\ &= \frac{(N+3)(N+2)(N+1)N!}{3!N!} \\ &= \frac{(N+3)(N+2)(N+1)}{6} \\ &= \frac{N^3+6N^2+11N+6}{6}. \end{aligned} \tag{3.4.1}$$

It may be helpful at this point to restate some information about $\tilde{C}_4(N)$ that was first stated in Section 2.2.5. This expression represents the number of unique ways that an N-document collection can be split into 4 mutually exclusive categories such that the sum of the category cardinalities is always equal to N (the total number of documents in the collection). The cardinality of each of these categories is an integer in the closed interval $[0, N]$.

The number, $\tilde{C}_4(N)$, when calculated for a particular value of N, say 100, is the total number of weak compositions that a combinatorial enumeration algorithm would have to examine to determine how many of them had a respective p value (the proportion of relevant documents that contained the query term) that was greater than their respective t value (the proportion of all documents that contained the query term).

Based on the formulation of Equation 3.4.1, $\tilde{C}_4(N)$ is $\Theta(N^3)$(Graham et al., 1994); that is, it has a cubic growth rate and the bound is tight. Table 3.9 on the following page illustrates how rapidly this function's values grow as the value of its input parameter increases. Roughly speaking, doubling the size of its input causes its output to change by a factor of 8. It helps to demonstrate why brute force combinatorial techniques, while tractable for very small collection sizes, becomes increasingly intractable as the collection size scales up. Even a document collection size of merely 1,000 may tax the memory and processor resources that are associated with many personal computers, because the number of weak compositions that is associated with a collection of this size is over 167 million, according to the table below.

Table 3.9: Number of Weak Compositions of Size 4 for Selected Values of N.

N	$\tilde{C}_4(N)$
4	35
10	286
20	1,771
50	23,426
100	176,851
500	21,084,251
1,000	167,668,501

3.5 The Three Research Questions

This dissertation provided answers to the three research questions that are detailed below in the next three subsections. Each of these questions starts with an introduction that is immediately followed by a discussion of a sequence of actions that, when followed, provided the answer, or answers, to the question in a later chapter.

3.5.1 What would be the characteristics of a combinatoric measure, based on the ASL, that performs the same as a probabilistic measure of retrieval performance, also based on the ASL?

This question was answered by performing the following sequence of actions: (1) define the parameters of a combinatoric model that can be used to characterize the following ranking methods: best-case, worst-case, random case, inverse document frequency, decision-theoretic, and coordination level matching; (2) define each ranking method-specific model in terms of these parameters; (3) determine a formula to compute the number of events of interest for each model; and (4) develop a formula that computes the total number of events that can occur in each model. Next, use the results from the four steps above to develop combinatoric formulas for the normalized search length($\mathcal{A}$) (Losee, 1998) in an optimal ranking and the quality of a ranking ($\mathcal{Q}$) (Losee, 1998). In particular, the ranking method-specific formula for $\mathcal{Q}$ would be its formula corresponding to (3) divided by its formula corresponding to (4). Then use the formulas for $\mathcal{A}$ and $\mathcal{Q}$ to develop the ranking-specific formulas for the various ASL measures. Note that the expressions for $\mathcal{A}$ and $\mathcal{Q}$ are independent variables, so to speak, with respect to the formulas for the ASL. The formula for the ASL in Equation 3.5.1 and the equations for the $\mathcal{Q}$ measures in Table 3.10 on the following page are from Losee (1998).

$$\text{ASL} = N\left(\mathcal{Q}\mathcal{A} + \overline{\mathcal{Q}}\,\overline{\mathcal{A}}\right) + 1/2 \tag{3.5.1}$$

Note that $\overline{\mathcal{Q}} = 1 - \mathcal{Q}$ and $\overline{\mathcal{A}} = 1 - \mathcal{A}$. Both $\mathcal{Q}$ and $\mathcal{A}$ are real-valued entities in the range $[0, 1]$.

Finally, the last step in this process was the development of test data and strategies to help validate several of the formulas that were developed above. Each test scenario consisted of data, the formulas that were being tested, and the expected results from applying those formulas. For the smaller datasets, the results were able to be calculated manually. For the larger ones, a combination of manual calculations and programmatic calculations by Mathematica (Wolfram, 2003) were used. NOTE: The data created in this phase was used in the Data Analysis phase to help with the validation of the formulas developed above.

Table 3.10: Comparing Quality of Ranking Methods.

Ranking Method	$\mathcal{Q}$ (the degree of optimality)		
Best-case	$\mathcal{Q}_{\text{BC}}$	$=$	$\Pr(p > t, p > t) + \Pr(p \leq t, p \leq t)$
		$=$	1
Random	$\mathcal{Q}_{\text{RNDM}}$	$=$	$(\Pr(p > t) + \Pr(p \leq t))/2$
		$=$	$1/2$
Worst-case	$\mathcal{Q}_{\text{WC}}$	$=$	$\Pr(p > t, p \leq t) + \Pr(p \leq t, p > t)$
		$=$	0
Decision-theoretic	$\mathcal{Q}_{\text{DT}}$	$=$	$\Pr(p > t, p > q) + \Pr(p \leq t, p \leq q)$
		$=$	$\Pr(p > \max(t, q)) + \Pr(p \leq \min(t, q))$
Inverse Document Frequency	$\mathcal{Q}_{\text{IDF}}$	$=$	$\Pr(p > t, t > 0) + \Pr(p \leq t, t \leq 0)$
		$=$	$\Pr(p > t)$
Coordination Level Matching	$\mathcal{Q}_{\text{CLM}}$	$=$	$\Pr(p > t)$

An Example of How to Compute the ASL for Specified N, $\mathcal{Q}$, and $\mathcal{A}$ Values

Assume that for a 10 document collection and a single-term query, the ranking method-specific formulas, developed for this research, were used to calculate $\mathcal{A} = 0.75$, $\mathcal{Q} = 0.9$, and

$$\begin{aligned} \text{ASL} &= N\left(\mathcal{Q}\mathcal{A} + \overline{\mathcal{Q}}\;\overline{\mathcal{A}}\right) + 1/2 \\ &= N\left(\mathcal{Q}\mathcal{A} + (1 - \mathcal{Q})(1 - \mathcal{A})\right) + 1/2 \\ &= 10\ (0.9 * 0.75 + (1 - 0.9) * (1 - 0.75)) + 0.5 \end{aligned}$$

$$= 7.5.$$

This indicates that the average position of a relevant document in the ordering is 7.5 documents from the front of the ranked list (which is worse than the mean rank for a relevant document if the ranking algorithm randomly ordered documents according to a uniform distribution), that the normalized position of a relevant document is 0.75 (worse than average because the expected value would be 0.5 from an algorithm than does random ranking according to a uniform distribution), but that the quality of the ranking method is 0.9 (which is very good).

3.5.2 Does the ASL measure produce the same performance result as the result that would be obtained by a process that ranks documents and, then, calculates the Average Search Length from this empirical ranking data?

The question was answered by performing the sequence of actions described below. Develop computer software (e.g., Mathematica, Java (Flanagan, 2005), and/or C++ (Stroustrup, 2000) programs) that implements each of the 6 ranking algorithms that were mentioned near the beginning of Section 3.5.1. For each query in the CF′ collection: rank the documents in the collection by each of the 6 ranking algorithms; compute the *predicted* ASL value for each ranking method; compute the *actual* ASL value for each ranking method; and record this information in a dataset. This dataset has four columns: one for the query identifier, one for identifying the ranking algorithm, one for the predicted ASL value, and one for the actual ASL value.

An Example of How to Compute the ASL from A Strongly Ordered Ranking

Assume there is a ranked list of 10 documents for a particular query q and that each document has a distinct retrieval status value (RSV) (Table 3.11 depicts this situation). From the front to the back of the list, the documents are ranked 1 through 10, inclusive. Rank 1 is the best rank that a document can have, rank 10 is the worst one. In general, lower-numbered ranks are more desirable than higher-numbered ranks because the front of the list is defined to be rank 1.

Table 3.11: Ranked List of Ten Documents.

rank	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
relevant?	Yes	Yes	No	No	Yes	No	Yes	No	No	No
term present?	Yes	No	No	Yes	No	Yes	Yes	Yes	No	Yes

For this query/ranking method/document set combination, we would compute

$$\text{ASL} = (1 + 2 + 5 + 7)/4 = 3.75.$$

This would be the "actual" ASL value for this combination. To calculate the "predicted" ASL value for this combination, we would use the formula for $\mathcal{A}$ and the ranking method-specific formulas for ASL and $\mathcal{Q}$ that are developed in Chapters 4 through 7, inclusive.

In order to use these formulas, especially the one for $\mathcal{Q}$ (the quality of a ranking method), we need to calculate the model parameters. These are r_0 (the number of relevant documents where the query term is not present), r_1 (the number of relevant documents where the term is present), s_0 (the number of non-relevant documents where the query term is not present), and s_1 (the number of non-relevant documents where the term is present). For the data in Table 3.11, those values would be: $r_0 = 2$ (because only documents 2 and 5 meet the criteria for this category), $r_1 = 2$ (because only documents 1 and 7 meet the criteria for this category), $s_0 = 2$ (because only documents 3 and 9 meet the criteria for this category), and $s_1 = 4$ (because only documents 4, 6, 8, and

10 meet the criteria for this category). These values can be plugged into the method-specific formulas to calculate ASL, $\mathcal{Q}$, and $\mathcal{A}$. The ASL value is the "predicted" one and is recorded in the dataset (that is used later in the analysis phase for this RQ) along with the "actual" ASL (just computed from the above data), the query identifier, and the ranking method identifier.

3.5.3 When does the ASL measure and one of these measures (i.e., MZE, ESL, and MRR) both imply that one document ranking is better than another document ranking?

To answer this question, the following two actions were performed. Descriptions of these actions are detailed thusly.

The first action was to develop a way to compare the performance between measures that are "based on the totality of the search process" (Losee, 2000) (e.g., ASL) as well as those measures that "determine performance at a point in the search process" (Losee, 2000) (e.g., MZE, ESL).

The second action was to generate graphs that compared CM_ASL (the combinatoric version of the ASL performance measure) against the MZE, ESL, and RR performance measures. Note that, for a single query, the MRR and RR performance measures always yield identical results. We used Measure A and Measure B to denote the measures that are being compared. The items of interest were the regions of the graphs where 1 of the 2 measures, say Measure A, indicated that performance was either increasing, or was staying the same, within a region R, with respect to document positions in a ranked collection of documents, while performance, according to Measure B, was increasing within region R.

With respect to the first action, the work in Losee (2000) was extended to handle the requirements of that action. The Losee work developed techniques for comparing the differences between several measures (e.g., measure theory-based E measure (MZE), mean reciprocal rank (MRR), expected search length (ESL)) that compared performance at arbitrary points in the search process. The research for this dissertations extended that work. In this dissertation research, the mixture of measures were heterogeneous in nature (i.e., some the measures mentioned in this research question are "point" measures whereas others were "totality" measures). In Chapter 10 (The ASL Measure and Three Frequently-Used Performance Measures), the ASL, ESL, MZE, and RR measures were extended so that they were point measures whose calculated values were consistent with the assumption that *some* of the documents in a vector V of ranked documents *may* have tied (i.e., duplicate) RSVs.

3.6 Summary

This chapter discussed a strategy for accomplishing the stated research goals of this dissertation. It introduced the test collections that this research used and other resources such as the PubMed stopword list, the Cystic Fibrosis test collection, and the Porter stemmer. It discussed the reason that the Cystic Fibrosis test collection was not in a form that was appropriate for its intended use in this research and outlined a procedure to create an adapted version of it that could be used in this research. The detailed account of how to accomplish the adaptation is located in Appendix A.

In addition to the discussion that is in the first paragraph of this summary, this chapter provided more detailed statements on the three research questions and more details on what this research intended to accomplish, particularly with the calculation of the quality of ranking measures. A small example was provided that showed how the quality of ranking measure could be calculated for the CLM ranking method.

Chapter 4

Characteristics of a Combinatoric-Based Quality of Ranking Measure

If the number of terms used in the title of this chapter was required to be restricted to solely one term, "counting" would be an excellent choice for the title because the overall purpose of this chapter is to describe how to count some of the outcomes in the sample space for the combination of an information need, a document collection, and a ranking method. This chapter discusses characteristics of Average Search Length (ASL)-related performance models that are factors in the the development of expressions that compute the cardinality of certain parts of this sample space. In particular, this chapter shows how the sample space can be divided into 4 parts to make the counting process easier, details the effects that singularities can have on the counts, provides a solution that handles these singularities, and concludes with specific expressions that compute the cardinalities of subsets of the sample space that meet specified restrictions.

This chapter derives many equations and tables for cardinality counting. The major product of the work in this chapter is Table 4.11 on page 140, at the end of this chapter. Table 4.11 contains information on the number of outcomes, for certain constraints, of the sample space of weak 4-compositions for an N-document collection. This number of outcomes information is used in Chapter 5 and Chapter 6 to help with the derivation of equations that calculate the quality of ranking for the coordination level matching (CLM), decision-theoretic (DT), and inverse document frequency (IDF) ranking methods.

In turn, these quality of ranking equations from Chapters 5 and 6, are used in Section 7.8 and Section 7.10 to develop equations for the normalized and unnormalized search lengths, along with equations for the expected value and variance of these search lengths. These equations from Chapters 5, 6, and 7 also occupy a prominent role in Chapter 8 during the validation of formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ measures.

The counting for some subsets of the sample space is too complex to discuss in this general chapter. The counting for these subsets is handled in the more specialized chapters for the CLM, IDF, and DT ranking methods. These two specialized chapters, namely, Chapter 5 and Chapter 6, immediately follow this one.

4.1 Essential Characteristics

What were the essential characteristics of the ASL-based performance models that enabled us to better understand (and predict) the behavior of the CLM, IDF, and DT performance measures? The essential entities were a document collection, a set of information needs (realized by a set of queries), the performance measures themselves, and a set of parameters that were induced by various combinations of queries and the document collection.

From the document collection perspective, the most basic piece of information was N, the cardinality of the collection. Two essential characteristics of the models were binary relevance and that the single query term was either present or absent in a document. Since a document could be relevant, or not, and the term may, or may not, be present, only 4 variables were needed to represent the number of documents for the $2 \times 2 = 4$

possible categories. These variables were r_1 (the number of relevant documents where the query term was present), r_0 (the number of relevant documents where the query term was absent), s_1 (the number of non-relevant documents where the query term was present), and s_0 (the number of non-relevant documents where the query term was absent).

The count of relevant documents for a query in a model was represented by the expression $r_1 + r_0$ and the count of non-relevant documents was represented by the expression $s_1 + s_0$. These sums, in conjunction with various conditions on their values, are represented by Figure 4.1 on page 115. This figure not only gives a hint of the strategy that is used later to partition the sample space (its cardinality is $\tilde{C}_4(N)$), in order to make various calculations easier and to aid conceptual understanding, but it also references parameters p (the probability that the query term is present in a relevant document), t (the probability that an arbitrary document in the collection contains the query term), and q (the probability that the query term is present in a non-relevant document). These three parameters were directly derived from the more basic parameters $r_1,\ r_0,\ s_1,$ and s_0 (shown by Expressions 3.3.4 to 3.3.6 on page 91). Table 3.10 on page 105 shows that the quality of ranking measures for CLM and IDF are defined in terms of p and t and that the analogous measure for DT is defined in terms of p, q, and t.

Each of these parameters may be undefined for some events in the sample space. For example, as long as the document collection is non-empty, t is always defined. However, depending on the values of $r_1,\ r_0,\ s_1,$ and s_0, the values of p and q may be defined at times, and undefined at others, due to a singularity (e.g., a denominator that has a value of 0). The information in Figure 4.1 on page 115 includes the conditions under which $p,\ t,$ and q are either defined or not defined.

Another area of concern was that, even in situations where $p,\ t,$ and q were all defined, a particular quality of ranking measure may be undefined. The reason for this was that various values of these parameters led to situations where a document weighting function was undefined because the denominator in one of more parts of its defining expression was 0, or the entire expression evaluated to 0 (and was used as input to a logarithmic function). This situation and the ones in the previous paragraphs are discussed in more detail below.

4.2 The Document Collection Sample Space and Its Division Into Four Quadrants

The definitions of the quality of the ranking measures for an N document collection, and specific ranking methods, are given below along in terms of the parameters p, t, and q. Definitions of these parameters, as expressions involving the counts $r_1,\ r_0,\ s_1,$ and s_0, also appear below.

For any collection of $N \geq 0$ documents, there are several quantities that had to be calculated in order to be able to determine the $\mathcal{Q}$ measures for the CLM, IDF, and DT ranking methods. Their respective equations are

$$\begin{aligned}\mathcal{Q}_{\text{CLM}} &= \Pr(p > t),\\ \mathcal{Q}_{\text{IDF}} &= \Pr(p > t) + \Pr(p \leq t, t = 1),\ \text{and}\\ \mathcal{Q}_{\text{DT}} &= \Pr(p > \max(t, q)) + \Pr(p \leq \min(t, q)).\end{aligned}$$

The quantities that needed to be calculated were the size of the sample space (i.e., number of weak 4-compositions of size N); the number of outcomes where $p > t$; the number of outcomes where $p > t$ or the joint conditions $p \leq t$ and $t = 1$ hold; the number of outcomes where $p > \max(t, q)$; and the number of outcomes where $p \leq \min(t, q)$. Being able to calculate the latter 4 quantities, of course, was dependent on being able to calculate p, t, and q for each outcome in the sample space.

For the convenience of the reader, we restate the following information from page 90:

$$\begin{aligned}
p &= \frac{r_1}{r_0 + r_1}, \\
t &= \frac{r_1 + s_1}{r_0 + r_1 + s_0 + s_1}, \text{ and} \\
q &= \frac{s_1}{s_0 + s_1}.
\end{aligned}$$

Figure 4.1 on the following page divides the sample space into 4 quadrants that are based on the values of the expressions $r_1 + r_0$ and $s_1 + s_0$. In each of these quadrants, there is information that specifies whether each of p, t, and q is defined or undefined. The number of outcomes is also specified for each quadrant.

In information retrieval (IR) terminology, a *sample space* for a weak-composition of size k, and N documents, represents all the possible collections of N documents in terms of the k parameters. For example, $k = 4$ in many of the discussions in this chapter and subsequent ones. When $k = 4$, the parameters are r_1, r_0, s_1, and s_0. An *outcome* is an element of this sample space and represents exactly one of its collections.

The fourth nonblank line in each quadrant in Figure 4.1 on the next page represents the number of weak 4-compositions which fall into that quadrant in a collection with $N \geq 1$ documents. This number is 0 for Quadrant II because it is impossible for both the numbers of relevant and non-relevant documents to simultaneously be 0 when N is positive. If the number of relevant documents is 0 (i.e., $r_0 + r_1 = 0$), and the number of non-relevant documents is positive (i.e., $s_0 + s_1 > 0$), which is the situation in Quadrant III, then the number of weak 4-compositions is $\tilde{C}_2(N)$. Similarly, if the number of non-relevant documents is 0 (i.e., $s_0 + s_1 = 0$) and the number of relevant documents is positive, which is the situation in Quadrant I, then the number of weak 4-compositions for that quadrant is also $\tilde{C}_2(N)$. Both of these situations correspond to weak 4-compositions with a fixed value of 0 for two of its components. Effectively, this means that we are

		$r_0 + r_1$		
		0	> 0	
$s_0 + s_1$	0	p is undefined II q is undefined t is undefined 0	p is defined I q is undefined t is defined $\tilde{C}_2(N)$	$\tilde{C}_2(N)$
	> 0	p is undefined III q is defined t is defined $\tilde{C}_2(N)$	p is defined IV q is defined t is defined $\tilde{C}_4(N) - 2\tilde{C}_2(N)$	$\tilde{C}_4(N) - \tilde{C}_2(N)$
		$\tilde{C}_2(N)$	$\tilde{C}_4(N) - \tilde{C}_2(N)$	$\tilde{C}_4(N)$

Figure 4.1: The various quadrants illustrate the conditions under which p, q, and t are defined/undefined. The fourth row in each quadrant represents the number of outcomes in the sample space that meet the conditions for that particular quadrant for a collection of $N \geq 1$ documents. Note that, for positive N, the events in Quadrant II cannot occur; hence, the number of weak 4-compositions in that quadrant is 0.

interested in how many weak 2-compositions there are for N. There are $\tilde{C}_2(N)$ of them. Quadrant IV corresponds to those weak 4-compositions where there are at least two documents, with at least one of them being relevant and at least one of them being non-relevant. Since any weak 4-composition of N is associated with exactly one quadrant and the grand total for the quadrants must sum to $\tilde{C}_4(N)$, the number of weak 4-compositions for Quadrant IV is $\tilde{C}_4(N) - 2\tilde{C}_2(N)$.

To be complete, we need to cover the case where N can also have the value 0, that is, $N = 0$. The parameters of interest (i.e., p, q, t) are undefined in each of the 4 quadrants when $N = 0$. There is only one quadrant (i.e., Quadrant II) where it is possible to have a valid weak 4-composition when the collection of documents is empty. The only weak 4-composition that can occur is $(0, 0, 0, 0)$, thereby meaning that the count for Quadrant II is 1. Quadrants I, III, and IV represent impossibilities because their respective joint conditions that are a function of $r_1 + r_0$ and $s_1 + s_0$ cannot be true because at least one of the 4 values in any weak 4-composition for those quadrants must be a positive integer. Since it is impossible, for an empty document collection, to construct any weak 4-composition that satisfies the membership conditions for the latter three quadrants, the counts for Quadrants I, III, and IV must be 0.

The mathematical singularities that arose in some of the computations for p and q in Quadrants I and III of Figure 4.1 on the preceding page posed problems for our formulas that determined counts for the inverse document frequency, coordination level matching, and decision-theoretic document weighting functions because each of those functions were at least partially dependent on parameters p and t. In Quadrant I, a singularity was present for each possible value of q because the number of non-relevant documents was 0, thereby meaning that the denominator in the formula for q was also 0. Similarly, in Quadrant III, a singularity was present for each possible value of p because the number of relevant documents was 0, thereby meaning that the denominator in the formula for p was also 0.

Even in Quadrant IV, where p, t, and q were always defined for any weak 4-composition that was a member of that quadrant, there were singularities that had to be taken into account for the decision-theoretic document weighting function. This was due to the DT weighting function being defined as

$$\log\left(\frac{p/(1-p)}{q/(1-q)}\right). \tag{4.2.1}$$

Similarities occurred in this function when either of the following was true: $p = 0$, $p = 1$, $q = 0$, or $q = 1$. There were various ways to adapt the calculations for p, t, and q to eliminate possible singularities.

For the $N = 0$ case, the computation of each the p, q, and t values, for any of the 4 quadrants, was impossible due to singularities.

4.3 Handling Mathematical Singularities

In information retrieval, the typical way of handling potential singularities in document- or term-weighting functions is to modify, or adapt, the formulas in these functions so that singularities are impossible when the adapted versions of those functions are used to calculate the weights.

The basic document weighting functions (Losee, 1998; Salton and Buckley, 1988) for the CLM, IDF, and DT rankings are, respectively, *any* positive number w; $-\log(t)$; and $\log\left(\frac{p/(1-p)}{q/(1-q)}\right)$. This research assumed that the document weight for any CLM ranking was always $w = 1$.

Earlier, it was mentioned that the decision-theoretic weighting function that was used in this dissertation was based on binary independent features. The classic (i.e., conventional) way to adapt a weighting function to handle singularities has been to add a small positive integer c to the value in each of the cells of Figure 2.2 on page 25 so that the modified formulas for p and q always have positive denominators and so that no other singularities can occur in the DT weighting function. After this adaptation, we have

$$\begin{aligned} p &= \frac{r_1 + c}{r_1 + r_0 + 2c}, \\ q &= \frac{s_1 + c}{s_1 + s_0 + 2c}, \text{ and} \\ t &= \frac{r_1 + s_1 + 2c}{r_1 + r_0 + s_1 + s_0 + 4c}. \end{aligned}$$

When $c = 0.5$, we have the classic (i.e., conventional) adaptation and, hence,

$$\begin{aligned} p &= \frac{r_1 + 0.5}{r_1 + r_0 + 1}, && (4.3.1) \\ q &= \frac{s_1 + 0.5}{s_1 + s_0 + 1}, \text{ and} && (4.3.2) \\ t &= \frac{r_1 + s_1 + 1}{r_1 + r_0 + s_1 + s_0 + 2}. \end{aligned}$$

Some of the problems with this adaptation are that Equations 4.3.1 and 4.3.2 are not unbiased estimators of p and q (Shaw, 1995). These equations can overestimate both p and q when the number of relevant documents is small (van Rijsbergen et al., 1981; Yu et al., 1983). These equations can also overestimate both p and q when the number of relevant documents is large (Shaw, 1995). In addition, the "conventional computing equations can produce illogical outcomes when $c = 0.5$ dominates the computation of p" (Shaw, 1995).

One alternative to setting c to one-half (i.e., $c = 0.5$) is to set it to 1 (i.e., $c = 1$) (de Vries and Roelleke, 2005). This setting possesses the same problems that the classic, or conventional, setting of 0.5 possesses. The difference between the two for the contingency table of Figure 2.2 on page 25 is that instead of that table having two "virtual documents" (de Vries and Roelleke, 2005) added to it with the $c = 0.5$ setting, it has 4 with the $c = 1$ setting. The advantage of this setting is that the value in each cell of the contingency table is now a whole number. This, conceptually, makes it easier to interpret the values in each cell of the table as representing a number of documents (real plus virtual), rather than a number of documents and some fractional adjustment factor.

Another alternative is to set $c = n_1/N$ (Robertson, 1986). This "can be expected to resolve the problems of undefined and over estimated values of [Equation 4.2.1 on page 117], in most cases" (Shaw, 1995). Shaw cautioned, however, that singularities could still be present in certain situations where the document collection was small and its members were subject-related.

In addition, Shaw felt that "[i]t [was] unnecessary and inappropriate, however, to modify *all* [emphasis is that of the dissertation author] calculations of defining equations for [p] and [q] to resolve isolated mathematical singularities" (Shaw, 1995). He proposed a set of equations for p and q in which the singularities were handled as special cases. If $p = 0$ or $r_1 + r_0 = 0$, set $p = 1/N^2$. If $q = 0$ or $s_1 + s_0 = 0$, set $q = 1/N^2$. If $p = 1$ or $q = 1$, then set $p = 1 - 1/N^2$ or $q = 1 - 1/N^2$, respectively. Shaw (1995) states that "[t]he square of collection size insures that probabilities of magnitude 0 are reasonably estimated in a small set of retrieved documents or a small test collection." Shaw continued by noting that these modifications "alter the defining equations only as needed and resolve previously described computational difficulties."

After incorporating Shaw's proposals and extending them so that they include an empty document collection (i.e., $N = 0$), and a collection with a single document (i.e., $N = 1$), the result was

$$p' = \begin{cases} p, & \text{if } 0 < p < 1; \\ 10^{-4}, & \text{if } N \leq 1 \text{ and } (r_1 = 0 \text{ or } r_1 + r_0 = 0); \\ 1 - 10^{-4}, & \text{if } N = 1 \text{ and } r_1 = 1; \\ \frac{1}{N^2}, & \text{if } N \geq 2 \text{ and } (r_1 = 0 \text{ or } r_1 + r_0 = 0); \\ 1 - \frac{1}{N^2}, & \text{if } N \geq 2 \text{ and } r_1 \geq 1 \text{ and } r_0 = 0. \end{cases} \tag{4.3.3}$$

and

$$q' = \begin{cases} q, & \text{if } 0 < q < 1; \\ 10^{-4}, & \text{if } N \leq 1 \text{ and } (s_1 = 0 \text{ or } s_1 + s_0 = 0); \\ 1 - 10^{-4}, & \text{if } N = 1 \text{ and } s_1 = 1; \\ \frac{1}{N^2}, & \text{if } N \geq 2 \text{ and } (s_1 = 0 \text{ or } s_1 + s_0 = 0); \\ 1 - \frac{1}{N^2}, & \text{if } N \geq 2 \text{ and } s_1 \geq 1 \text{ and } s_0 = 0. \end{cases} \tag{4.3.4}$$

Shaw (1995) does not directly address the computation of t. The formula below is an extension of that work and illustrates how to calculate t'.

$$t' = \begin{cases} t, & \text{if } 0 < t < 1; \\ 10^{-4}, & \text{if } N \leq 1 \text{ and } (r_1 + s_1 = 0 \text{ or } N = 0); \\ 1 - 10^{-4}, & \text{if } N = 1 \text{ and } r_1 + s_1 = 1; \\ \frac{1}{N^2}, & \text{if } N \geq 2 \text{ and } r_1 + s_1 = 0; \\ 1 - \frac{1}{N^2}, & \text{if } N \geq 2 \text{ and } r_1 + s_1 = N. \end{cases} \tag{4.3.5}$$

The expressions denoted by p', t', and q' replaced those denoted by p, t, and q, respectively, in the formulas for $\mathcal{Q}_{\text{CLM}}$, $\mathcal{Q}_{\text{IDF}}$, and $\mathcal{Q}_{\text{DT}}$ that appeared on the first page of this chapter.

Therefore, the equations at the beginning of this chapter were rewritten as

$$\begin{aligned}
\mathcal{Q}'_{\text{CLM}} &= \Pr(p' > t'), \\
\mathcal{Q}'_{\text{IDF}} &= \Pr(p' > t') + \Pr(p' \leq t', t' = 1 - \epsilon) \\
&= \mathcal{Q}'_{\text{CLM}} + \Pr(p' \leq t', t' = 1 - \epsilon), \text{ and} \\
\mathcal{Q}'_{\text{DT}} &= \Pr(p' > \max(t', q')) + \Pr(p' \leq \min(t', q'))
\end{aligned}$$

where

$$\epsilon = \begin{cases} N^{-2}, & \text{if } N \geq 2; \\ 10^{-4}, & \text{otherwise.} \end{cases}$$

The use of p', q', and t' in place of their original counterparts also affected the formulas for the normalized search length ($\mathcal{A}$) and unnormalized search length (ASL) measures. Their analogous redefinitions were

$$\mathcal{A}' = (1 - p' + t')/2$$

and

$$\text{ASL}' = N(\mathcal{Q}'\mathcal{A}' + (1 - \mathcal{Q}')(1 - \mathcal{A}')) + 1/2,$$

respectively, where $\mathcal{Q}'$ was one of $\mathcal{Q}'_{\text{CLM}}, \mathcal{Q}'_{\text{DT}},$ or $\mathcal{Q}'_{\text{IDF}}$. Further discussion and use of the $\mathcal{Q}', \mathcal{A}'$, and ASL$'$ formulas take place in Chapters 7 and 8.

4.4 What and Why Do We Count?

For each ranking method, the objective was to count the number of qualifying outcomes for that method for a given N (the number of documents in a collection). An outcome was said to be *qualifying*, for a ranking method, if its p', t', and q' values satisfied the conditions for this method. For example, only those outcomes with p' and t' values, such that the condition $p' > t'$ held, were qualifying ones for $\mathcal{Q}_{\text{CLM}} = \Pr(p' > t')$. The $\mathcal{Q}$ value for a particular method was calculated by dividing the count for the method by the number of weak 4-compositions corresponding to N.

Fundamental to this counting process was the calculation of the p', t', and q' values for each outcome. These values were used to help determine whether the outcome was a qualifying one. In order to make these calculations more manageable, separate analyses were performed for each of the quadrants in Figure 4.1 on page 115. Near the end of this chapter, the results of these analyses were combined and placed in Table 4.11 on page 140.

Near the end of each of the analyses in the discussions to follow, the counts are presented as a 3-tuple (that is denoted as a *count contribution triple*). The components of such a triple, from the first component to the last one, are, respectively, the number of outcomes where the condition $p' > t'$ holds, the number of outcomes where the condition $p' \leq \min(t', q')$ holds, and the number of outcomes where the condition $p' > \max(t', q')$ holds.

Within an analysis, a count contribution triple is only valid under certain conditions. In particular, one of the major requirements is that the size of a document collection must equal or exceed a specified threshold in order for the triple to be applicable within the quadrant that it is associated with. This threshold varies by the characteristics of a quadrant; it is exactly 0 in Quadrant II (i.e., N=0), 1 in Quadrants I and III (i.e.,

$N \geq 1$), and ranges from 2 to 4, inclusive, in Quadrant IV, depending on the subcondition for a particular count contribution triple. More is written about this in the various analyses below. Each analysis details how the size condition is determined for its count contribution triple(s).

Regardless of the number of documents that are in a collection, Quadrants I, II, and III only have the potential of contributing very small amounts to the count for any ranking method. In fact, their contribution potential is almost insignificant for large N. This is because their proportion of the total number of outcomes can be proved to monotonically decrease as the collection size increases. By contrast, Quadrant IV is where the overwhelming bulk of the contributions come from for the CLM, IDF, and DT ranking methods. Except when both p and q are in the open interval $(0, 1)$, closed form solutions can be obtained for all of the cases discussed for each quadrant in the remainder of this chapter. These cases show how to derive the closed form solutions.

4.5 Determining the Number of Qualifying Document Collections for Quadrant I (each weak 4-composition in this quadrant represents a document collection that has at least one relevant document and zero non-relevant documents)

This is the first of several sections that develop equations to calculate the contribution counts for their respective quadrants. The equations that are developed in these sections are used in subsequent chapters to develop equations that calculate the quality of ranking values for the coordination level matching, inverse document frequency, and decision-theoretic methods.

The event that corresponds to this subset of the sample space has $\tilde{C}_2(N) = N + 1$ outcomes because each outcome has zero non-relevant documents (i.e., $s_1+s_0 = 0$) that is associated with it and there are only $N+1$ distinct ways in which the weak 2-compositions that correspond to the positive number of relevant documents (i.e., $r_1 + r_0 > 0$) can be constructed. Table 4.1 on the next page illustrates this and other relationships. Being in this quadrant implies that the cardinality of the document collection is at least one (i.e., $N \geq 1$) because the smallest value that the sum $r_1 + r_0$ can have is 1.

Since all the documents are relevant in each outcome of this event, due to there not being any non-relevant documents in this quadrant, the p' value for an outcome is the same as the associated t' value for that outcome. This can be easily proved by first noting that $r_1 + r_0 > 0$ implies that either (1) $r_1 = 0$ and $r_0 > 0$; (2) $r_1 > 0$ and $r_0 = 0$; or (3) $r_1 > 0$ and $r_0 > 0$. In addition, $s_1 + s_0 = 0$ implies that $s_1 = s_0 = 0$ because s_1 and s_0 are both natural numbers.

When $r_1 = 0$ and $r_0 > 0$, both t and p are equal to 0 which means that $t' = p' = 10^{-4}$, if $N = 1$; and that $t' = p' = 1/N^2$, otherwise. When $r_1 > 0$ and $r_0 = 0$, both t and p are equal to 1 which means that $t' = p' = 1 - 10^{-4}$, if $N = 1$; and that $t' = p' = 1/N^2$, otherwise. Finally, when $r_1 > 0$ and $r_0 > 0$, both t and p have the value $r_1/(r_1 + r_0)$. Since this value is in the open interval $(0, 1)$, and by the formulas (i.e., Equation 4.3.3 on page 120 and Equation 4.3.3 on page 120) for p' and t', we can assert that $p' = p = t' = t$. So, it follows that in each of the outcomes in this event $p' = t'$. The number of outcomes where $p' > t'$ is 0 because we just established that $p' = t'$ holds for all outcomes in this quadrant.

Since $s_1 + s_0 = 0$ is always true for each outcome in this quadrant, the calculation of q for each outcome leads to a singularity for each of these calculations. Therefore, $q' = 10^{-4}$, when $N = 1$, and $q' = 1/N^2$, otherwise.

Visual inspection of Table 4.1 on the following page indicates that $p' = t' \geq q'$ holds when $N \geq 1$. The number of outcomes for which $p' > \max(t', q')$ is 0. For either $N = 1$ or $N \geq 2$, the number of outcomes for which $p' \leq \min(t', q')$ is 1 and the number for which $p' > \max(t', q')$ is 0. Hence, the count contribution triple is $(0, 1, 0)$ when $N \geq 1$.

Table 4.1: Quadrant I Outcomes.

condition	r_1	r_0	s_1	s_0	p'	t'	q'
$N = 1$	0	1	0	0	10^{-4}	10^{-4}	10^{-4}
$N = 1$	1	0	0	0	$1 - 10^{-4}$	$1 - 10^{-4}$	10^{-4}
$N \geq 2$	0	N	0	0	$1/N^2$	$1/N^2$	$1/N^2$
$N \geq 2$	1	$N - 1$	0	0	$1/N$	$1/N$	$1/N^2$
$N \geq 2$	2	$N - 2$	0	0	$2/N$	$2/N$	$1/N^2$
$N \geq 2$	$\cdots$	$\cdots$	0	0	$\cdots$	$\cdots$	$1/N^2$
$N \geq 2$	$N - 1$	1	0	0	$(N - 1)/N$	$(N - 1)/N$	$1/N^2$
$N \geq 2$	N	0	0	0	$1 - (1/N^2)$	$1 - (1/N^2)$	$1/N^2$

4.6 Determining the Number of Qualifying Document Collections for Quadrant II (the single weak 4-composition in this quadrant represents the empty collection of documents for $N = 0$)

If $N = 0$, the event that corresponds to this subset of the sample space has one outcome, namely, $(0, 0, 0, 0)$ associated with it.

On the other hand, if $N \geq 1$, then the event that corresponds to this subset of the sample space has no outcomes associated with it.

Hence, the count contribution triple is $(0, 1, 0)$, if $N = 0$; otherwise, it is $(0, 0, 0)$.

4.7 Determining the Number of Qualifying Document Collections for Quadrant III (each weak 4-composition in this quadrant represents a document collection that has zero non-relevant documents and at least one relevant document)

The structure of the analysis for this quadrant is very similar to that for Quadrant I. The main difference is that we are using q and q' in those places where we used p and p', respectively, in the analysis for Quadrant I.

The event that corresponds to this subset of the sample space has $\tilde{C}_2(N) = N + 1$ outcomes, the same number as was present in Quadrant I. This is because each outcome has zero relevant documents (i.e., $r_1 + r_0 = 0$) associated with it and there are only $N + 1$ distinct ways in which the weak 2-compositions corresponding to the positive number of non-relevant documents (i.e., $s_1 + s_0 > 0$) can be constructed. Table 4.2 on the next page illustrates this and other relationships. Being in this quadrant implies that the cardinality of the document collection is at least 1 (i.e., $N \geq 1$).

All the documents are non-relevant in each outcome of this event, due to there not being any relevant documents in this quadrant. The q' value for an outcome is the same as the associated t' value for that outcome. This can be easily proved by first noting that $s_1 + s_0 > 0$ implies that either (1) $s_1 = 0$ and $s_0 > 0$; (2) $s_1 > 0$ and $s_0 = 0$; or (3) $s_1 > 0$ and $s_0 > 0$. In addition, $r_1 + r_0 = 0$ implies that $r_1 = r_0 = 0$ because r_1 and r_0 are both natural numbers.

When $s_1 = 0$ and $s_0 > 0$, both t and q are equal to 0 which means that $t' = q' = 10^{-4}$, if $N = 1$, and that $t' = q' = 1/N^2$, otherwise. When $s_1 > 0$ and $s_0 = 0$, both t and q are equal to 1 which means that $t' = q' = 1 - 10^{-4}$, if $N = 1$, and that $t' = q' = 1/N^2$, otherwise. Finally, when $s_1 > 0$ and $s_0 > 0$, both t and q have the value $s_1/(s_1 + s_0)$. Since this value is in the open interval $(0, 1)$, and by the formulas that start on page 120 for q' and t', we can assert that $q' = q = t' = t$. So, it follows that, in each of the outcomes in this event, $q' = t'$. The number of outcomes where $q' > t'$ is 0 because we just established that $q' = t'$ holds for all outcomes in this quadrant.

Since $r_1 + r_0 = 0$ is always true for each outcome in this quadrant, the calculation of p for each outcome leads to a singularity for each of the outcomes. Therefore, $p' = 10^{-4}$ when $N = 1$ and $p' = 1/N^2$, otherwise.

Visual inspection of Table 4.2 indicates that $p' \leq t' = q'$ holds when $N \geq 1$. The number of outcomes for which $p' > \max(t', q')$ is 0. For either $N = 1$ or $N \geq 2$, the number of outcomes for which $p' \leq \min(t', q')$ is $N + 1$ and the number for which $p' > \max(t', q')$ is 0. The count contribution triple is $(0, N + 1, 0)$ when $N \geq 1$.

Table 4.2: Quadrant III Outcomes.

condition	r_1	r_0	s_1	s_0	p'	t'	q'
$N = 1$	0	0	0	1	10^{-4}	10^{-4}	10^{-4}
$N = 1$	0	0	1	0	10^{-4}	$1 - 10^{-4}$	$1 - 10^{-4}$
$N \geq 2$	0	0	0	N	$1/N^2$	$1/N^2$	$1/N^2$
$N \geq 2$	0	0	1	$N - 1$	$1/N^2$	$1/N$	$1/N$
$N \geq 2$	0	0	2	$N - 2$	$1/N^2$	$2/N$	$2/N$
$N \geq 2$	0	0	$\cdots$	$\cdots$	$1/N^2$	$\cdots$	$\cdots$
$N \geq 2$	0	0	$N - 1$	1	$1/N^2$	$(N - 1)/N$	$(N - 1)/N$
$N \geq 2$	0	0	N	0	$1/N^2$	$1 - (1/N^2)$	$1 - (1/N^2)$

4.8 Determining the Number of Qualifying Document Collections for Quadrant IV (each weak 4-composition in this quadrant represents a document collection that has at least one relevant document and at least one non-relevant document)

Quadrant IV is, by far, the most complex quadrant to analyze. The analyses for it use values of p and q to divide the work into 9 mutually exclusive joint categories. A value for either p or q is in exactly one of three single categories: it is equal to 0, it is in the open interval $(0, 1)$, or it is equal to 1. Since the categories can be independently chosen for each of p and q, the result is a total of $3 \times 3 = 9$ mutually exclusive joint categories.

Contribution counts when Quadrant IV, $p = 0$, and $q = 0$. None of the documents contain the query term.

The event that corresponds to this subset of the sample space has $C_2(N) = N - 1$ outcomes because each outcome has zero documents with feature frequency 1 associated with it and there are only $N-1$ distinct ways in which the 2-compositions that correspond to the r_0 and s_0 components can be constructed. Table 4.3 on the next page illustrates this and other relationships. The first row in the table shows that because the value of $N - 1$ must be positive, then N must have a value of at least 2 (i.e., $N \geq 2$).

None of the outcomes of this event have any documents with feature frequency 1, the p', t', and q' values is the same in each outcome. This can be easily proved by noting that

$$
\begin{aligned}
p &= \frac{r_1}{r_1 + r_0} = \frac{0}{0 + r_0} = \frac{0}{r_0} = 0, \\
t &= \frac{r_1 + s_1}{r_1 + r_0 + s_1 + s_0} = \frac{0 + 0}{0 + r_0 + 0 + s_0} = \frac{0}{r_0 + s_0} = 0, \text{ and} \\
q &= \frac{s_1}{s_1 + s_0} = \frac{0}{0 + s_0} = \frac{0}{s_0} = 0
\end{aligned}
$$

in this context. It follows that, in each of the outcomes in this event, $p' = t' = q' = 1/N^2$. This allows us to state that $p' = t' = q'$ holds when $N \geq 2$, that the number of outcomes where $p' > t'$ is 0; that the number of outcomes for which $p' \leq \min(t', q')$ is $N - 1$; and that the number of outcomes for which $p' > \max(t', q')$ is 0. Hence, the count contribution triple is $(0, N - 1, 0)$.

Table 4.3: Quadrant IV Outcomes ($p = 0$ and $q = 0$ and $N \geq 2$).

r_1	r_0	s_1	s_0	p'	t'	q'
0	1	0	$N-1$	$1/N^2$	$1/N^2$	$1/N^2$
0	2	0	$N-2$	$1/N^2$	$1/N^2$	$1/N^2$
0	$\cdots$	0	$\cdots$	$1/N^2$	$1/N^2$	$1/N^2$
0	$N-2$	0	2	$1/N^2$	$1/N^2$	$1/N^2$
0	$N-1$	0	1	$1/N^2$	$1/N^2$	$1/N^2$

Contribution counts when Quadrant IV, $p = 0$, and $q \in (0, 1)$. None of the relevant documents contain the query term; some, but not all, of the non-relevant documents contain the query term.

The event that corresponds to this subset of the sample space has $C_3(N) = \binom{N-1}{2}$ outcomes because the number of relevant documents with feature frequency 1 is zero and the numbers of documents in each of the remaining three categories must be positive. Table 4.4 on page 131 illustrates this and other relationships. From the first row in the table we can see that because the value of $N - 2$ must be positive, then N must have a value of at least three (i.e., $N \geq 3$).

Since all of the relevant documents have feature frequency 0, $p' = 1/N^2$ in each outcome. Since there are no relevant documents with feature frequency 1 in each outcome of this event, the t' and q' values, with respect to an outcome, have the same numerator, before any simplification, because

$$\begin{aligned} t &= \frac{r_1 + s_1}{r_1 + r_0 + s_1 + s_0} = \frac{0 + s_1}{0 + r_0 + s_1 + s_0} = \frac{s_1}{r_0 + s_1 + s_0} = 0 \text{ and} \\ q &= \frac{s_1}{s_1 + s_0} \end{aligned}$$

in this context. Due to $r_0 > 0$ being true for each outcome, it follows that, in each outcome, $t' < q'$ is true, too. We also note that with our constraint of $N \geq 2$, that $p' < t'$ is true, too. Therefore, $p' < t' < q'$ also holds.

This allows us to state that the number of outcomes where $p' > t'$ is 0; that the number of outcomes for which $p' \leq \min(t', q')$ is $\binom{N-1}{2}$; and that the number of outcomes for which $p' > \max(t', q')$ is 0.

Hence, the count contribution triple is $\left(0, \binom{N-1}{2}, 0\right)$.

Contribution counts when Quadrant IV, $p = 0$, and $q = 1$. None of the relevant documents contain the query term; all of the non-relevant documents contain the query term.

The event that corresponds to this subset of the sample space has $C_2(N) = N - 1$ outcomes. This is because each outcome has zero relevant documents with feature frequency 1, has zero non-relevant documents with feature frequency 0 associated with it, and there are only $N - 1$ distinct ways in which the 2-compositions that correspond to the r_0 and s_1 components can be constructed. Table 4.5 on the next page illustrates this and other relationships. The first row in the table shows that because the value of $N - 1$ must be positive, then N must have a value of at least 2 (i.e., $N \geq 2$).

The above allows us to state that each outcome has $p' = 1/N^2$ and $q' = 1 - (1/N^2)$

Table 4.4: Quadrant IV Outcomes ($p = 0$ and $q \in (0, 1)$ and $N \geq 3$).

r_1	r_0	s_1	s_0	p'	t'	q'
0	1	1	$N-2$	$1/N^2$	$1/N$	$1/(N-1)$
0	1	2	$N-3$	$1/N^2$	$2/N$	$2/(N-1)$
0	1	3	$N-4$	$1/N^2$	$3/N$	$3/(N-1)$
0	1	$\cdots$	$\cdots$	$1/N^2$	$\cdots$	$\cdots$
0	1	$N-3$	2	$1/N^2$	$(N-3)/N$	$(N-3)/(N-1)$
0	1	$N-2$	1	$1/N^2$	$(N-2)/N$	$(N-2)/(N-1)$
0	2	1	$N-3$	$1/N^2$	$1/N$	$1/(N-2)$
0	2	2	$N-4$	$1/N^2$	$2/N$	$2/(N-2)$
0	2	$\cdots$	$\cdots$	$1/N^2$	$\cdots$	$\cdots$
0	2	$N-4$	2	$1/N^2$	$(N-4)/N$	$(N-4)/(N-2)$
0	2	$N-3$	1	$1/N^2$	$(N-3)/N$	$(N-3)/(N-2)$
0	$\cdots$	$\cdots$	$\cdots$	$1/N^2$	$\cdots$	$\cdots$
0	$N-3$	1	2	$1/N^2$	$1/N$	$1/3$
0	$N-3$	2	1	$1/N^2$	$2/N$	$2/3$
0	$N-2$	1	1	$1/N^2$	$1/N$	$1/2$

associated with it. The t' value for each outcome can be calculated by the expression s_1/N. Since the value of r_0 ranges from 1 to $N-1$, inclusive, for the outcomes in this event, and that value of r_0/N is always greater than $1/N^2$ and is always less than $1-(1/N^2))$, we can conclude that $p' < t' < q'$.

Hence, the count contribution triple is $(0, N-1, 0)$.

Table 4.5: Quadrant IV Outcomes ($p = 0$ and $q = 1$ and $N \geq 2$).

r_1	r_0	s_1	s_0	p'	t'	q'
0	1	$N-1$	0	$1/N^2$	$(N-1)/N$	$1-(1/N^2)$
0	2	$N-2$	0	$1/N^2$	$(N-2)/N$	$1-(1/N^2)$
0	$\cdots$	$\cdots$	$\cdots$	$1/N^2$	$\cdots$	$1-(1/N^2)$
0	$N-2$	2	0	$1/N^2$	$2/N$	$1-(1/N^2)$
0	$N-1$	1	0	$1/N^2$	$1/N$	$1-(1/N^2)$

Contribution counts when Quadrant IV, $p \in (0, 1)$, and $q = 0$. Some, but not all, of the relevant documents contain the query term; none of the non-relevant documents contain the query term.

The event that corresponds to this subset of the sample space has $C_3(N) = \binom{N-1}{2}$ outcomes because the number of non-relevant documents with feature frequency 1 is 0 and the numbers of documents in each of the remaining three categories must be positive. Table 4.6 on the following page illustrates this and other relationships. From the first row in the table, we can see that because the value of $N-2$ must be positive, then N must have a value of at least three (i.e., $N \geq 3$).

Since the number of non-relevant documents with feature frequency 1 is 0 in each outcome of this event, we have

$$\begin{aligned} p &= \frac{r_1}{r_1 + r_0}, \\ t &= \frac{r_1 + s_1}{r_1 + r_0 + s_1 + s_0} = \frac{r_1 + 0}{r_1 + r_0 + 0 + s_0} = \frac{r_1}{r_1 + r_0 + s_0}, \text{ and} \\ q' &= \frac{1}{N^2} \end{aligned}$$

in this context.

One of our items of interest is in discovering the relationship R (e.g., $=, \neq, >, \geq, <, \leq$) between p', t', and q'. We start by asserting that $p'Rt'$ is true for at least one of those 6 relational operators and writing

$$\begin{aligned} p'R\ t' &\equiv \frac{r_1}{r_1 + r_0}\ R\ \frac{r_1}{r_1 + r_0 + s_0} \\ &\equiv r_1(r_1 + r_0 + s_0)\ R\ (r_1 + r_0)r_1 \\ &\equiv r_1^2 + r_1 r_0 + r_1 s_0\ R\ r_1^2 + r_0 r_1 \\ &\equiv r_1 s_0\ R\ 0. \end{aligned}$$

Table 4.6 on the next page indicates that both r_1 and s_0 have positive values in each outcome. Therefore, R can be either the greater-than relationship or the greater-than-or-equal-to relationship. The more appropriate one to use in this situation is the greater-than relationship. Now, we can state that $p' > t'$. Since $t' > q'$ for all the outcomes when $N \geq 3$, we can also state that $p' > t' > q'$.

Hence, the count contribution triple is $\left(\binom{N-1}{2}, 0, \binom{N-1}{2}\right)$.

Table 4.6: Quadrant IV Outcomes ($p \in (0,1)$ and $q = 0$ and $N \geq 3$).

r_1	r_0	s_1	s_0	p'	t'	q'
1	1	0	$N-2$	$1/2$	$1/N$	$1/N^2$
1	2	0	$N-3$	$1/3$	$1/N$	$1/N^2$
1	3	0	$N-4$	$1/4$	$1/N$	$1/N^2$
$\cdots$	$\cdots$	0	$\cdots$	$\cdots$	$\cdots$	$1/N^2$
1	$N-3$	0	2	$1/(N-2)$	$1/N$	$1/N^2$
1	$N-2$	0	1	$1/(N-1)$	$1/N$	$1/N^2$
2	1	0	$N-3$	$2/3$	$2/N$	$1/N^2$
2	2	0	$N-4$	$2/4$	$2/N$	$1/N^2$
2	$\cdots$	0	$\cdots$	$\cdots$	$2/N$	$1/N^2$
2	$N-4$	0	2	$2/(N-2)$	$2/N$	$1/N^2$
2	$N-3$	0	1	$2/(N-1)$	$2/N$	$1/N^2$
$\cdots$	$\cdots$	0	$\cdots$	$\cdots$	$\cdots$	$1/N^2$
$N-3$	1	0	2	$(N-3)/(N-2)$	$(N-3)/N$	$1/N^2$
$N-3$	2	0	1	$(N-3)/(N-1)$	$(N-3)/N$	$1/N^2$
$N-2$	1	0	1	$(N-2)/(N-1)$	$(N-2)/N$	$1/N^2$

Contribution counts when Quadrant IV, $p \in (0,1)$, and $q \in (0,1)$. Some, but not all, of both the relevant and non-relevant documents contain the query term.

The computations of the counts for $p' > t'$, $p' \leq \min(t', q')$, and $p' > \max(t', q')$ are discussed in Chapter 5 and Section 6.3.

Contribution counts when Quadrant IV, $p \in (0,1)$, and $q = 1$. Some, but not all, of the relevant documents contain the query term; all of the non-relevant documents contain the query term.

The event that corresponds to this subset of the sample space has $C_3(N) = \binom{N-1}{2}$ outcomes because the number of non-relevant documents with feature frequency 1 is 0 and the numbers of documents in each of the remaining three categories must be positive. Table 4.7 on the following page illustrates this and other relationships. From the first row in the table, we can see that because the value of $N - 2$ must be positive, then N must have a value of at least three (i.e., $N \geq 3$).

Since the number of non-relevant documents with feature frequency 0 is zero in each outcome of this event, we have

$$p = \frac{r_1}{r_1 + r_0},$$
$$t = \frac{r_1 + s_1}{r_1 + r_0 + s_1 + s_0} = \frac{r_1 + s_1}{r_1 + r_0 + s_1 + 0} = \frac{r_1 + s_1}{r_1 + r_0 + s_1}, \text{ and}$$
$$q' = 1 - \frac{1}{N^2}$$

in this context.

One item of interest is in discovering the relationship R (e.g., $=, \neq, >, \geq, <, \leq$) between p', t', and q'. We start by asserting that $p'Rt'$ is true for at least one of those 6 relational operators and writing

$$\begin{aligned} p'R\, t' &= \frac{r_1}{r_1 + r_0} \; R \; \frac{r_1 + s_1}{r_1 + r_0 + s_1} \\ &= r_1(r_1 + r_0 + s_1) \; R \; (r_1 + r_0)(r_1 + s_1) \\ &= r_1^2 + r_1 r_0 + r_1 s_1 \; R \; r_1^2 + r_1 s_1 + r_0 r_1 + r_0 s_1 \\ &= 0 \; R \; r_1 s_1. \end{aligned}$$

Table 4.7 on the next page indicates that both r_1 and s_1 have positive values in each outcome. Therefore, R can be either the less-than relationship or the less-than-or-equal-to relationship. The more appropriate one to use in this situation is the former. Now, we can state that $p' < t'$. Since $t' < q'$ for all the outcomes when $N \geq 3$, we can also state that $p' < t' < q'$.

Hence, the count contribution triple is $\big(0, \binom{N-1}{2}, 0\big)$.

Table 4.7: Quadrant IV Outcomes ($p \in (0, 1)$ and $q = 1$ and $N \geq 3$).

r_1	r_0	s_1	s_0	p'	t'	q'
1	1	$N-2$	0	$1/2$	$(N-1)/N$	$1-(1/N^2)$
1	2	$N-3$	0	$1/3$	$(N-2)/N$	$1-(1/N^2)$
1	3	$N-4$	0	$1/4$	$(N-3)/N$	$1-(1/N^2)$
$\cdots$	$\cdots$	$\cdots$	0	$\cdots$	$\cdots$	$1-(1/N^2)$
1	$N-3$	2	0	$1/(N-2)$	$3/N$	$1-(1/N^2)$
1	$N-2$	1	0	$1/(N-1)$	$2/N$	$1-(1/N^2)$
2	1	$N-3$	0	$2/3$	$(N-1)/N$	$1-(1/N^2)$
2	2	$N-4$	0	$2/4$	$(N-2)/N$	$1-(1/N^2)$
2	$\cdots$	$\cdots$	0	$\cdots$	$\cdots$	$1-(1/N^2)$
2	$N-4$	2	0	$2/(N-2)$	$4/N$	$1-(1/N^2)$
2	$N-3$	1	0	$2/(N-1)$	$3/N$	$1-(1/N^2)$
$\cdots$	$\cdots$	$\cdots$	0	$\cdots$	$\cdots$	$1-(1/N^2)$
$N-3$	1	2	0	$(N-3)/(N-2)$	$(N-1)/N$	$1-(1/N^2)$
$N-3$	2	1	0	$(N-3)/(N-1)$	$(N-2)/N$	$1-(1/N^2)$
$N-2$	1	1	0	$(N-2)/(N-1)$	$(N-1)/N$	$1-(1/N^2)$

Contribution counts when Quadrant IV, $p = 1$, and $q = 0$. All of the relevant documents contain the query term; none of the non-relevant documents contain the query term.

The event that corresponds to this subset of the sample space has $C_2(N) = N - 1$ outcomes because each outcome has zero relevant documents with feature frequency 0 associated and zero non-relevant documents with feature frequency 1 associated with it. There are only $N - 1$ distinct ways in which the 2-compositions that correspond to the r_1 and s_0 components can be constructed. Table 4.8 on the following page illustrates this and other relationships. The first row in the table shows that because the value of $N - 1$ must be positive, then N must have a value of at least 2 (i.e., $N \geq 2$).

The information in the first paragraph of this case allows us to state that

$$\begin{aligned}
p &= \frac{r_1}{r_1 + r_0} = \frac{r_1}{r_1 + 0} = \frac{r_1}{r_1} = 1, \\
t &= \frac{r_1 + s_1}{r_1 + r_0 + s_1 + s_0} = \frac{r_1 + 0}{r_1 + 0 + 0 + s_0} = \frac{r_1}{r_1 + s_0}, \text{ and} \\
q &= \frac{s_1}{s_1 + s_0} = \frac{0}{0 + s_0} = \frac{0}{s_0} = 0
\end{aligned}$$

in this context. We can see right away that $p > t > q$ thereby letting us state that $p' > t' > q'$. Also, from these equations, we can state that $p' = 1 - (1/N^2)$ and $q' = 1/N^2$ in each outcome.

Hence, the count contribution triple is $(N - 1, 0, N - 1)$.

Table 4.8: Quadrant IV Outcomes ($p = 1$ and $q = 0$ and $N \geq 2$).

r_1	r_0	s_1	s_0	p'	t'	q'
1	0	0	$N - 1$	$1 - (1/N^2)$	$1/N$	$1/N^2$
2	0	0	$N - 2$	$1 - (1/N^2)$	$2/N$	$1/N^2$
$\cdots$	0	0	$\cdots$	$1 - (1/N^2)$	$\cdots$	$1/N^2$
$N - 2$	0	0	2	$1 - (1/N^2)$	$(N - 2)/N$	$1/N^2$
$N - 1$	0	0	1	$1 - (1/N^2)$	$(N - 1)/N$	$1/N^2$

Contribution counts when Quadrant IV, $p = 1$, and $q \in (0, 1)$. All of the relevant documents contain the query term; some, but not all, of the non-relevant documents contain the query term.

The event that corresponds to this subset of the sample space has $C_3(N) = \binom{N-1}{2}$ outcomes because the number of relevant documents with feature frequency 0 is zero and the numbers of documents in each of the remaining three categories must be positive. Table 4.9 on page 138 illustrates this and other relationships. The first row in the table shows that because the value of $N - 2$ must be positive, then N must have a value of at least three (i.e., $N \geq 3$).

The information in the first paragraph of this case allows us to state that

$$\begin{aligned}
p &= \frac{r_1}{r_1 + r_0} = \frac{r_1}{r_1 + 0} = \frac{r_1}{r_1} = 1, \\
t &= \frac{r_1 + s_1}{r_1 + r_0 + s_1 + s_0} = \frac{r_1 + s_1}{r_1 + 0 + s_1 + s_0} = \frac{r_1 + s_1}{r_1 + s_1 + s_0}, \text{ and} \\
q &= \frac{s_1}{s_1 + s_0}
\end{aligned}$$

in this context.

One of our items of interest is in discovering the relationship R (e.g., $=, \neq, >, \geq, <, \leq$) between p', t', and q'. We start by asserting that $t' R\ q'$ is true for at least 1 of those 6 relational operators. This assertion leads to the derivation of the following equivalence:

$$\begin{aligned}
t'\ R\ q' &\equiv \frac{r_1 + s_1}{r_1 + s_1 + s_0}\ R\ \frac{s_1}{s_1 + s_0} \\
&\equiv ((r_1 + s_1)(s_1 + s_0))\ R\ ((r_1 + s_1 + s_0)s_1) \\
&\equiv (r_1 s_1 + r_1 s_0 + s_1^2 + s_1 s_0)\ R\ (r_1 s_1 + s_1^2 + s_0 s_1) \\
&\equiv (r_1 s_0)\ R\ 0.
\end{aligned}$$

Table 4.9 on the following page indicates that both r_1 and s_0 have positive values in each outcome. Therefore, R can be either the greater-than relationship or the greater-than-or-equal-to relationship. The more appropriate one to use in this situation is the former. Now, we can state that $t' > q'$. Since $p' < t'$ for all the outcomes when $N \geq 3$, we can also state that $p' > t' > q'$. Hence, the count contribution triple is $\left(\binom{N-1}{2}, 0, \binom{N-1}{2}\right)$.

Contribution counts when Quadrant IV, $p = 1$, and $q = 1$. All of the documents contain the query term.

The event that corresponds to this subset of the sample space has $C_2(N) = N - 1$ outcomes because each outcome does not have any documents with feature frequency 1

Table 4.9: Quadrant IV Outcomes ($p = 1$ and $q \in (0, 1)$ and $N \geq 3$).

r_1	r_0	s_1	s_0	p'	t'	q'
1	0	1	$N-2$	$1-(1/N^2)$	$2/N$	$1/(N-1)$
1	0	2	$N-3$	$1-(1/N^2)$	$3/N$	$2/(N-1)$
1	0	3	$N-4$	$1-(1/N^2)$	$4/N$	$3/(N-1)$
1	0	$\cdots$	$\cdots$	$1-(1/N^2)$	$\cdots$	$\cdots$
1	0	$N-3$	2	$1-(1/N^2)$	$(N-2)/N$	$(N-3)/(N-1)$
1	0	$N-2$	1	$1-(1/N^2)$	$(N-1)/N$	$(N-2)/(N-1)$
2	0	1	$N-3$	$1-(1/N^2)$	$3/N$	$1/(N-2)$
2	0	2	$N-4$	$1-(1/N^2)$	$4/N$	$2/(N-2)$
2	0	$\cdots$	$\cdots$	$1-(1/N^2)$	$\cdots$	$\cdots$
2	0	$N-4$	2	$1-(1/N^2)$	$(N-2)/N$	$(N-4)/(N-2)$
2	0	$N-3$	1	$1-(1/N^2)$	$(N-1)/N$	$(N-3)/(N-2)$
$\cdots$	0	$\cdots$	$\cdots$	$1-(1/N^2)$	$\cdots$	$\cdots$
$N-3$	0	1	2	$1-(1/N^2)$	$(N-2)/N$	$1/3$
$N-3$	0	2	1	$1-(1/N^2)$	$(N-1)/N$	$2/3$
$N-2$	0	1	1	$1-(1/N^2)$	$(N-1)/N$	$1/2$

associated with it. There are only $N - 1$ distinct ways in which the 2-compositions that correspond to the r_1 and s_1 components can be constructed. Table 4.10 on the next page illustrates this and other relationships. The first row in the table shows that because the value of $N - 1$ must be positive, then N must have a value of at least 2 (i.e., $N \geq 2$).

The information in the first paragraph of this case allows us to state that

$$\begin{aligned}
p &= \frac{r_1}{r_1 + r_0} = \frac{r_1}{r_1 + 0} = \frac{r_1}{r_1} = 1, \\
t &= \frac{r_1 + s_1}{r_1 + r_0 + s_1 + s_0} = \frac{r_1 + s_1}{r_1 + 0 + s_1 + 0} = \frac{r_1 + s_1}{r_1 + s_1} = 1, \text{ and} \\
q &= \frac{s_1}{s_1 + s_0} = \frac{s_1}{s_1 + 0} = \frac{s_1}{s_1} = 1
\end{aligned}$$

in this context. We can see right away that $p = t = q$, thereby letting us state that $p' = t' = q'$. Also, from these equations, we can state that $p' = t' = q' = 1 - (1/N^2)$ in each outcome.

Hence, the count contribution triple is $(0, N - 1, 0)$.

Table 4.10: Quadrant IV Outcomes ($p = 1$ and $q = 1$ and $N \geq 2$).

r_1	r_0	s_1	s_0	p'	t'	q'
1	0	$N-1$	0	$1-(1/N^2)$	$1-(1/N^2)$	$1-(1/N^2)$
2	0	$N-2$	0	$1-(1/N^2)$	$1-(1/N^2)$	$1-(1/N^2)$
$\cdots$	0	$\cdots$	0	$1-(1/N^2)$	$1-(1/N^2)$	$1-(1/N^2)$
$N-2$	0	2	0	$1-(1/N^2)$	$1-(1/N^2)$	$1-(1/N^2)$
$N-1$	0	1	0	$1-(1/N^2)$	$1-(1/N^2)$	$1-(1/N^2)$

4.9 Summary

The various contribution count results that were obtained from the preceding discussions for Quadrants I, II, III, and IV were consolidated in Table 4.11 on the following page. This information is used in Chapter 5 (A Combinatoric Model of $\mathcal{Q}'$ for the CLM Ranking Method) and Chapter 6 (A Combinatoric Model of $\mathcal{Q}'$ for the IDF and DT Ranking Methods) to calculate the quality of ranking measures for the methods of interest in these chapters.

Table 4.11: Number of Outcomes for the Four Quadrants (lines 4–12, inclusive, represent Quadrant IV.)

LINE	QUAD	size condition	supplemental condition	# of outcomes satisfying $p' > t'$	# of outcomes satisfying $p' \leq \min(t', q')$	# of outcomes satisfying $p' > \max(t', q')$
1	I	$N \geq 1$		0	1	0
2	II	$N = 0$		0	1	0
3	III	$N \geq 1$		0	$N + 1$	0
4	IV	$N \geq 2$	$p = 0$ and $q = 0$	0	$N - 1$	0
5		$N \geq 3$	$p = 0$ and $q \in (0, 1)$	0	$\binom{N-1}{2}$	0
6		$N \geq 2$	$p = 0$ and $q = 1$	0	$N - 1$	0
7		$N \geq 3$	$p \in (0, 1)$ and $q = 0$	$\binom{N-1}{2}$	0	$\binom{N-1}{2}$
8		$N \geq 4$	$p \in (0, 1)$ and $q \in (0, 1)$	derived in Chapter 5	derived in Chapter 6	derived in Chapter 6
9		$N \geq 3$	$p \in (0, 1)$ and $q = 1$	0	$\binom{N-1}{2}$	0
10		$N \geq 2$	$p = 1$ and $q = 0$	$N - 1$	0	$N - 1$
11		$N \geq 3$	$p = 1$ and $q \in (0, 1)$	$\binom{N-1}{2}$	0	$\binom{N-1}{2}$
12		$N \geq 2$	$p = 1$ and $q = 1$	0	$N - 1$	0

Chapter 5

A Combinatoric Model of $\mathcal{Q}'$ for the Coordination Level Matching Ranking Method

The purpose of this chapter is to develop counting expressions that collectively calculate the quality of the coordination level matching (CLM) ranking method for a document collection of size N. Some of these expressions come from the general work that was discussed in Chapter 4. The work in this chapter, along with that in Chapter 6, enable us to calculate the ranking method-specific $\mathcal{Q}'$ values that are referenced in many of the equations that are in Section 7.10 (A Family of ASL Measures), which starts on page 327, and Section 8.2 (The Validation of $\mathcal{Q}'$ Estimates That Were Obtained by Random Sampling), which starts on page 348.

The CLM quality of ranking equation, that is derived later in this chapter, is used in Section 7.8 and Section 7.10 of Chapter 7 to help develop equations for the normalized and unnormalized search lengths, along with equations for the expected value and variance of these search lengths. This equation also occupies a prominent role in Chapter 8 during the validation of formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ measures.

This chapter is the first of two consecutive chapters that derive expressions to help calculate the quality of a specific ranking method. The next chapter does this for the inverse document frequency (IDF) and decision-theoretic (DT) ranking methods. Many ideas and concepts are introduced and developed in this chapter; its material is largely based on combinatorial arguments and mathematical proofs. Each of these chapters have a short section that describes the respective ranking method of interest and the derivation of the quality of ranking measure that is associated with it.

It is usually the norm in the kind of research being undertaken by this dissertation that any assertion be either formally proved or disproved. Throughout this chapter, and in the other chapters of this dissertation, there are many assertions that emerge from parts of this research. Almost invariably, these assertions have a lemma or theorem associated with them. In this research, formal proofs were provided for these lemmas and theorems.

For the convenience of the reader, we restate the following concepts from Chapter 2 and Section 4.2. From the information retrieval (IR) perspective of this dissertation, these concepts cover the notions of weak composition, composition, and sample space.

A weak composition of size 4 represents a collection of N documents where *at least one* of the following conditions must be true: the number of relevant documents that contain the query term is 0 (i.e., $r_1 = 0$), the number of relevant documents that do not contain the query term is 0 (i.e., $r_0 = 0$), the number of non-relevant documents that contain the query term is 0 (i.e., $s_1 = 0$), or the number of non-relevant documents that do not contain the query term is 0 (i.e., $s_0 = 0$).

A strong composition of size 4 represents a collection of N documents where *all* of the following conditions must be true: the number of relevant documents that contain the query term is positive (i.e., $r_1 \geq 1$), the number of relevant documents that do not contain the query term is positive (i.e., $r_0 \geq 1$), the number of non-relevant documents that contain the query term is positive (i.e., $s_1 \geq 1$), and the number of non-relevant documents that do not contain the query term is positive (i.e., $s_0 \geq 1$).

A sample space for a weak-composition of size k, and N documents, represents all the possible collections of N documents in terms of the k parameters. For example, $k = 4$ in many of the discussions in this chapter and the discussions in subsequent chapters. When $k = 4$, the parameters are r_1, r_0, s_1, and s_0. An *outcome* is an element of this sample space and represents exactly one of its collections.

On page 121 of Chapter 4, it was stated that the quality of the CLM ranking method is defined by the following equation:

$$\mathcal{Q}'_{\text{CLM}} = \Pr(p' > t'). \tag{5.0.1}$$

For $N \geq 0$, the value of $\mathcal{Q}'_{\text{CLM}}$ can be calculated by determining the count contribution of the number of weak 4-compositions in the sample space for an N document collection, when $p' > t'$, and then dividing this value by the cardinality of the sample space. The sample space of weak 4-compositions for a query represents all the possible ways that queries can induce the partitioning of the document collection into a set of weak 4-compositions.

In summary, a weak 4-composition for a document collection C of size N represents the cardinalities that are associated with a partitioning of C into four sets such that the union of these sets yields C. The four partitions correspond to the set of relevant documents that contain the query term, the set of relevant documents that do not contain the query term, the set of non-relevant documents that contain the query term, and the set of non-relevant documents that do not contain the query term. The associated cardinalities are represented by r_1, r_0, s_1, and s_0, respectively.

For a weak 4-composition, the number of documents in a partition can range from as few as zero documents to as many documents as there are in the entire collection (i.e., N). It is an invariant that, for any weak 4-composition of an N document collection, the sum of the values for the four parts of the composition is always equal to N. In a strong 4-composition, as contrasted with a weak 4-composition, everything that was just stated above about a weak 4-composition applies, but, in addition, each partition must contain at least one document and no partition can contain more than $N - 3$ documents.

5.1 Ranking By Coordination Level Matching

Coordination level matching, also known as simple matching in the IR literature, is one of the first techniques that researchers used to study the ranking of documents for a query or set of queries. Conceptually, for a query-document pair, CLM ranking first determines the distinct terms in both the query q and the document. Next, it calculates how many distinct terms the query q and document have in common. The resultant value is the retrieval status value (RSV) for the document. After this process has taken place for all the documents in the collection that are associated with query q, the documents are ranked by their RSVs. The ordering, that is induced by the ranking, places the documents with the higher magnitude RSVs at higher ranks than those documents that have the lower magnitude RSVs. If two or more documents have the same RSV, these documents appear consecutively in the ordering, but after those documents that have higher ranks and before those documents that have lower ranks. Note that ranks are represented by natural numbers that start at 1 and end at the number of documents N in a collection. The highest possible rank is 1, the next highest possible rank is 2, and so on, with the lowest possible rank being N.

Largely due to its simplicity and the way that CLM works, many documents in a collection may be assigned the same RSV. For example, consider a query q with the set of terms

$$\{\text{quick, brown, fox, jump, fence}\}.$$

The documents that contain all five of these terms have an RSV of 5, the ones that only have 4 of these terms have an RSV of 4, and so on. The documents that contain none of these terms have 0 as their RSV. Generally, there are likely to be more documents that do not contain all the query terms than there are documents that contain all of them. In CLM ranking, it is not at all unusual for there to be large numbers of duplicate RSVs among the documents. The weight of a document in CLM ranking is a constant $c > 0$.

5.2 Two Basic Ways to Count the Number of Qualifying Weak 4-Compositions

In IR terms, the calculation (or determination) of the number of weak 4-compositions for N that satisfy the condition $r_1 s_0 > r_0 s_1$ (analogous to the condition $p' > t'$) is equivalent to finding the number of collections, over all the possible collections of N documents, that satisfy the condition $r_1 s_0 > r_0 s_1$.

There are two basic ways that one can go about developing arithmetic expressions that compute the number of qualifying weak 4-compositions of N. One is direct, the other is indirect. The direct way concentrates on how to develop expressions for counting the number of weak 4-compositions of N that satisfy the restriction that is denoted by $r_1 s_0 > r_0 s_1$. Satisfaction of Constraints 3.3.12 to 3.3.14 on page 94 is implicit in the definition of 4-compositions of N. The indirect way makes use of the fact that

$$C_4(N) = \text{Card4C}(N, r_1 s_0 < r_0 s_1) + \text{Card4C}(N, r_1 s_0 = r_0 s_1) + \text{Card4C}(N, r_1 s_0 > r_0 s_1),$$

and that it is sometimes easier to calculate the number of weak 4-compositions that satisfy the restriction $r_1 s_0 = r_0 s_1$ than it is to compute the number that satisfy $r_1 s_0 > r_0 s_1$. The notation Card4C(N, *restriction*) denotes the number of 4-compositions of N after the set of 4-compositions of N has been subsetted by the condition that is denoted by *restriction.* If, for any N, we can prove that the number of weak 4-compositions of N that satisfy $r_1s_0 < r_0s_1$ is equal to the number of weak 4-compositions of N that satisfies $r_1s_0 > r_0s_1$, then the number of weak 4-compositions for N can be re-expressed as

$$C_4(N) = 2 \cdot \text{Card4C}(N, r_1s_0 > r_0s_1) + \text{Card4C}(N, r_1s_0 = r_0s_1).$$

This means that if we can compute the number of weak 4-compositions of N that satisfy $r_1s_0 = r_0s_1$, then we can determine the number of weak 4-compositions of N that satisfy $r_1s_0 > r_0s_1$ rather easily. The next lemma establishes that the number of weak 4-compositions that satisfy the condition $r_1s_0 > r_0s_1$ in an N document collection for a query q has the same value as the number of weak 4-compositions that satisfy the condition $r_1s_0 < r_0s_1$. This helps us to prove that once we have a way to determine the number of weak 4-compositions that satisfy $r_1s_0 = r_0s_1$, we also have a way to determine how many satisfy the condition $r_1s_0 > r_0s_1$. This fact is useful later on in this chapter.

Lemma 5.2.1. *The number of weak* 4*-compositions of* N *that satisfies* $r_1s_0 > r_0s_1$ *is equal to the number of weak* 4*-compositions of* N *that satisfies* $r_1s_0 < r_0s_1$.

Proof. Any weak 4-composition of N that satisfies $r_1s_0 > r_0s_1$ can be represented by a 4-tuple where the first component contains the value for r_1, the second component contains the value for s_0, the third component contains the value for r_0, and the last component contains the value for s_1. Let a, b, c, and d represent an instance of respective values for these components such that these values satisfy $r_1s_0 > r_0s_1$. For any particular instance of a 4-tuple, say, (a, b, c, d), that is in the set of weak 4-compositions of N that satisfy $r_1s_0 > r_0s_1$, the instances (a, b, d, c), (b, a, c, d), and (b, a, d, c) also satisfy this relation because the products of the values of the first 2 and last 2 components of each instance are $a \times b$ and $c \times d$, respectively.

Each of the four 4-tuple instances in the previous paragraph can be transformed easily into an instance that satisfies $r_1 s_0 < r_0 s_1$ by the following two actions: interchange the value of its first component with that of its third component, then interchange the value of its second component with that of its fourth component. This yields the 4-tuples (a, b, c, d), (a, b, d, c), (b, a, c, d), and (b, a, d, c), from the 4-tuples (c, d, a, b), (d, c, a, b), (c, d, b, a), and (d, c, b, a), respectively. By definition of the set of weak 4-compositions for N, the former set of 4-tuple instances are also members of this set. Furthermore, the cardinality of the former set is exactly the same as that of the latter set. □

5.3 The Number of Distinct 2-Partitions

Compositions, both weak and strong, are an integral part of the many formula derivations that occur later in this chapter. The two equations for determining the number of weak and strong compositions appear in Chapter 2, starting on page 26. In addition to needing these equations, we also need an equation that determines the number of 2-partitions for the derivations that occur later in Section 5.11.2 and Section 5.11.3 in this chapter.

Partitions and compositions are closely related. In fact, to a large degree, each of these mathematical structures can be defined in terms of the other one. A partition of a positive integer n is an unordered sum of positive integers that has the value n. By contrast, compositions are ordered sums of positive integers and weak compositions are ordered sums of natural numbers. By convention in mathematics, the value of partition parts are listed in non-increasing order. For example, when viewed as unordered sums, the seven partitions of the number 5 are:

$$5$$

$$4+1$$

$$3+2$$

$$\begin{aligned}&3+1+1\\&2+2+1\\&2+1+1+1\\&1+1+1+1+1.\end{aligned}$$

There is one partition with 1 part, two partitions with 2 parts, two partitions with 3 parts, one partition with 4 parts, and one partition with 5 parts.

In this dissertation, our interest was in the number of distinct 2-partitions of n. A 2-partition of n is a partition of n that has exactly 2 parts. The distinct 2-partitions of n can be obtained from the associated 2-partitions by simply discarding the, at most one, 2-partition where both parts have the same value. For example, the 2-partitions of the number 5 are:

$$\begin{aligned}&4+1\\&3+2.\end{aligned}$$

These are also the distinct 2-partitions of 5 because both parts of the sum have different values. As another example, consider the 2-partitions of the number 6:

$$\begin{aligned}&5+1\\&4+2\\&3+3.\end{aligned}$$

These partitions include one (i.e., 3+3) that has the same value for both of its parts. If we discard this partition, we obtain the two distinct 4-partitions of 6, that is,

$$5+1$$

$$4+2.$$

In partition theory, the partition function $Q(n,k)$ denotes the number of distinct k-partitions for the positive integer n. The general formula for this particular function is defined in terms of the partition function $P(n,k)$, which is recursive in nature, and denotes the number of k-partitions of n. The work in this chapter does not need the use of the general version of either $Q(n,k)$ or $P(n,k)$, which is good, because relatively simple, and closed form, versions of these functions only exist for very small (e.g., k=1,2,3,4) values of k. Based on the work in Comtet (1974), the partition function $Q(n,2)$, where $n \geq 1$, can be defined as:

$$Q(n,2) = \begin{cases} \frac{n-2}{2}, & \text{if } n \text{ is even;} \\ \frac{n-1}{2}, & \text{otherwise.} \end{cases}$$

With the use of the greatest integer function, it can be defined even more succinctly as

$$Q(n,2) = \left\lfloor \frac{n-1}{2} \right\rfloor.$$

5.4 Divisor Pairs and Prime Power Factorizations

A key part of the input to each algorithm that we develop, or use, in subsequent sections of this chapter is the value of N. The number of divisors that N has, and whether a collection of these divisors are *relatively prime* (Rosen et al., 2000; Rosen, 2005), are very important to the development of each algorithm. Two positive integers are relatively prime if and only if their greatest common divisor is 1. For example, the pairs 2 and 15; 8 and 9; and 27 and 52 are relatively prime because they have no factors in common other than the number 1. On the other hand, the pair 2 and 14 are not relatively prime because the number 2 is the largest integer that divides them. Informally, an integer a is said to *divide* another integer b when the value denoted by b/a can be expressed as an integer. For example, 2 divides 6 because 6/2=3. But, 2 does not divide 5 because the value denoted by 5/2 is 2.5 (which cannot be expressed by an integer).

Let N be a positive integer. *The Fundamental Theorem of Arithmetic* (Rosen et al., 2000; Rosen, 2005) states that any positive integer can be written in a unique way as a product of powers of increasing prime numbers. More precisely,

$$N = {p_1}^{a_1}{p_2}^{a_2}\dots{p_m}^{a_m}$$

where $p_1, p_2, \dots, p_m$ are the m unique primes in N,

$$2 \leq p_1 < p_2 < \cdots < p_m \leq \sqrt{N},$$

and a_i is the number of times, including zero, that p_i occurs in the written representation of N. Of course, it is assumed in the last sentence that i is an integer with a value that is between 1 and m, inclusive. The representation of N as a product of powers of primes is called a *prime-power factorization* (Rosen, 2005) of N. A primary interest in this dissertation was how many unique divisors a positive integer N has and the values of these divisors. The number of these divisors can be computed rather easily if one has the prime-power factorization of N. The number of unique divisors is simply the product of the values obtained by adding 1 to the exponent of each of the primes in that factorization. For example, the prime-power factorization of 12 is $2^2 \cdot 3^1$. Therefore, we expect that 12 has $(2 + 1)(1 + 1) = 6$ unique divisors. Indeed, the unique divisors of 12 are 1, 2, 3, 4, 6, and 12.

There are $\tau(N)$ *divisor pairs* of N. Note that the expression $\tau(N)$ is standard notation, used in the area of mathematics known as elementary number theory (Rosen, 2005), to denote the number of unique positive integer divisors of a positive integer N. In the set expression below, the divisors of N are assumed to be ascendingly ordered (i.e., the smallest divisor is denoted by $v_1^{(N)}$, the second smallest is denoted by $v_2^{(N)}$, the third smallest is denoted by $v_3^{(N)}$, and so on; the largest divisor is denoted by $v_{\tau(N)}^{(N)}$). By definition, the smallest divisor is always 1; the largest one is always N. The set of divisor pairs of N is

$$\{(v_1^{(N)}, N/v_1^{(N)}), (v_2^{(N)}, N/v_2^{(N)}), (v_3^{(N)}, N/v_3^{(N)}), \ldots, (v_{\tau(N)}^{(N)}, N/v_{\tau(N)}^{(N)})\},$$

where $v_1^{(N)} = 1$ and $v_{\tau(N)}^{(N)} = N$. A property that is possessed by each divisor pair is that the product of its two positive integer components is always equal to N. This property is very important to our subsequent analyses. The set of divisor pairs for 12 (shown by Table 5.1 on the following page) is

$$\{(1, 12), (2, 6), (3, 4), (4, 3), (6, 2), (12, 1)\}.$$

The set of divisor pairs for 16 is

$$\{(1, 16), (2, 8), (4, 4), (8, 2), (16, 1)\}.$$

For their respective Ns, each of these sets of divisor pairs represents all the possible ways that two ordered positive integers can be multiplied to yield the product that is equal to that particular N. Notationally, let $d_i^{(N)}$ denote the ith divisor pair for N; that is,

$$d_i^{(N)} = (v_i^{(N)}, N/v_i^{(N)}) = (v_i^{(N)}, v_{\tau(N)+1-i}^{(N)}).$$

For divisor pair $d_i^{(N)}$, let $d_i^{(N)}[1]$ represent the value of its first component (i.e., $v_i^{(N)}[1]$) and $d_i^{(N)}[2]$ represent the value of its second component (i.e., $N/v_i^{(N)}[1]$).

Table 5.1: Divisor Pair Mappings for $N = 12$.

$d_1^{(12)}$	$d_2^{(12)}$	$d_3^{(12)}$	$d_4^{(12)}$	$d_5^{(12)}$	$d_6^{(12)}$
$(1, 12)$	$(2, 6)$	$(3, 4)$	$(4, 3)$	$(6, 2)$	$(12, 1)$

5.5 Basic Divisor Pair-Related Definitions

Information from the following definitions are used throughout this chapter. Definitions 5.5.0.1 to 5.5.0.6 on pages 152–153, inclusive, are associated with important sets that are based on the concept of divisor pairs. The other two definitions map a divisor pair to either a set of generally qualifying strong 4-compositions (stated by Definition 5.5.0.7 on page 153) or to a set of generally qualifying weak 4-compositions (stated by Definition 5.5.0.8 on page 154). In subsequent sections of this dissertation, the term *composition,* used without the modifier *weak*, is synonymous with the term *strong composition.*

Definition 5.5.0.1. Let the 2-*tuple representation of the set of divisor pairs* for N be represented by the set

$$T^{(N)} = \{(a, b) \in \mathbb{Z}^+ \times \mathbb{Z}^+ | ab = N \text{ and } a, b \in \mathbb{Z}^+\}.$$

Definition 5.5.0.2. Let the 4-*tuple representation of the set of divisor pairs* for strong 4-compositions of N be represented by the set

$$D^{(N)} = \{(w, x, y, z) | (a, b) \in T^{(N)} \text{ and } \ w + x = a \text{ and } \ y + z = b \text{ and } w, x, y, z \in \mathbb{Z}^+\}.$$

Definition 5.5.0.3. Let the 4-*tuple representation of the set of divisor pairs* for weak 4-compositions of N be represented by the set

$$\tilde{D}^{(N)} = \{(w, x, y, z) | (a, b) \in T^{(N)} \text{ and } \ w + x = a \text{ and } \ y + z = b \text{ and } w, x, y, z \in \mathbb{N}\}.$$

Definition 5.5.0.4. Let the set of 4-tuples $D_i^{(N)}$, where $i \in [\tau(N)]$, represent the set of tuples associated with divisor pair $d_i^{(N)}$. This 4-*tuple representation of a divisor pair* $d_i^{(N)}$ for strong 4-compositions is the set

$$D_i^{(N)} = \{(w, x, y, z) \in D^{(N)} | w + x = d_i^{(N)}[1] \text{ and } \ y + z = d_i^{(N)}[2]\}.$$

Definition 5.5.0.5. Let the set of 4-tuples $\tilde{D}_i^{(N)}$, where $i \in [\tau(N)]$, represent the set of tuples associated with divisor pair $d_i^{(N)}$. This 4-*tuple representation of a divisor pair* $d_i^{(N)}$ for weak 4-compositions is the set

$$\tilde{D}_i^{(N)} = \{(w, x, y, z) \in \tilde{D}^{(N)} | w + x = d_i^{(N)}[1] \text{ and } \ y + z = d_i^{(N)}[2]\}.$$

Definition 5.5.0.6. The *divisor pair weak composition mapping function*

$$\text{dpwcm} : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$$

maps the 4-tuple from $D_i^{(N)}$ or $\tilde{D}_i^{(N)}$, where $i \in [\tau(N)]$, to a weak 4-composition where the product of its first 2 components is always equal to the product of its last 2 components. It is defined as

$$\text{dpwcm}(w, x, y, z) \to (wy, xz, wz, xy).$$

If the 4-tuple is from the set $D_i^{(N)}$, then the weak composition that it is mapped to is also a strong composition because each element of the weak composition is a positive integer.

Definition 5.5.0.7. Let the set of 4-tuples $G_i^{(N)}$, where $i \in [\tau(N)]$, represent the set of *generally qualifying* 4-*compositions* associated with divisor pair $d_i^{(N)}$. The 4-tuple generally qualifying strong composition representation of a divisor pair $d_i^{(N)}$ is the set

$$G_i^{(N)} = \{\text{dpwcm}(w, x, y, z) | (w, x, y, z) \in D_i^{(N)}\}.$$

Definition 5.5.0.8. Let the set of 4-tuples $\tilde{G}_i^{(N)}$, where $i \in [\tau(N)]$, represent the set of *generally qualifying weak* 4-*compositions* associated with divisor pair $d_i^{(N)}$. The 4-tuple generally qualifying weak composition representation of a divisor pair $d_i^{(N)}$ is the set

$$\tilde{G}_i^{(N)} = \{\text{dpwcm}(w, x, y, z) | (w, x, y, z) \in \tilde{D}_i^{(N)}\}.$$

5.6 Number-Theoretic-Based Fundamentals of a Solution

One of the major goals in this chapter was the derivation of equations that count the number of weak 4-compositions of N that satisfy $r_1 s_0 = r_0 s_1$. A weak 4-composition that satisfies this condition corresponds to a document collection where the condition $p' = t'$ is true. That is, the probability p' that a relevant document in the collection contains the query term is the same as the unconditional probability t' that any document in the collection contains the query term.

Once we are able to determine the number of weak 4-compositions of N that satisfy condition $r_1 s_0 = r_0 s_1$, we are able to easily determine the numbers of weak 4-compositions that satisfy conditions $r_1 s_0 < r_0 s_1$ and $r_1 s_0 > r_0 s_1$ by the use of Lemma 5.2.1 on page 146. After obtaining these numbers, the values of $\Pr(p' = t')$, $\Pr(p' < t')$, and $\Pr(p' > t')$, respectively, are trivial to determine. All we need to do, then, is to divide these numbers by $\tilde{C}_4(N)$, the number of weak 4-compositions of N. By Equation 5.0.1 on page 143, the value for the last of these probabilities (i.e., $\Pr(p' > t')$) is the same as the value for $\mathcal{Q}'_{\text{CLM}}$.

Conceptually, the task that we just discussed in the prior two paragraphs is simple to explain and understand. Basically, we count the number of qualifying weak 4-compositions and divide that number by a number that represents the cardinality of all the possible weak 4-compositions for N. However, while the task above is conceptually simple, the work to effect the counting is nontrivial. This is a characteristic that is shared by many enumeration problems.

> There is no rule which says that enumeration techniques, even the simplest one, must have solutions expressible as closed formulas. (Lovász, 2007)

Basically, this statement makes note of the fact that conceptually simple enumeration problems, like the one that is the focus of this chapter, often have nontrivial solutions and involve much work. The main enumeration problem that we solve in this chapter has a nontrivial solution that requires much effort to develop.

Part of the reason that Lovász's statement is germane to what occurs in this chapter is because the equations that are associated with the various conditions that appear in subsequent sections must have integer solutions. Equations of this type are known as *Diophantine* equations (Rosen, 2005), and are generally more difficult to solve than equations that can have real or complex number solutions. Another reason is that, with the rather general nature of some of these conditions, it is difficult, if not impossible, to predict whether one or more solutions even exist for an arbitrary positive value of N.

5.6.1 The General Constraints

The collection of *general constraints* that any solution must satisfy are

$$r_1 s_0 = r_0 s_1$$

$$r_0 + r_1 + s_0 + s_1 = N$$

$$\begin{aligned} r_0, r_1, s_0, s_1 &\in \mathbb{N} \\ N &\in \mathbb{Z}^+. \end{aligned} \tag{5.6.1}$$

The symbol $\mathbb{N}$ denotes the set of natural numbers and the expression $\mathbb{Z}^+$ denotes the set of positive integers. The set of weak 4-compositions that satisfy these four general constraints are said to be *generally qualifying* weak 4-compositions.

5.6.2 The Form of A Solution that Satisfies the General Constraints

We are interested in all assignments of values to variables r_1, s_0, r_0, and s_0 that satisfy the general constraints above. By making use of the Fundamental Theorem of Arithmetic (Rosen, 2005), and the concept of divisors of a positive integer N, we have

$$\begin{aligned} N &= r_1 + s_0 + r_0 + s_1 \\ &= d_i^{(N)}(N/d_i^{(N)}) \\ &= d_i^{(N)} d_{\tau(N)+1-i}^{(N)} \end{aligned}$$

where $i \in [\tau(N)]$ and $d_i^{(N)}$ is the ith divisor for N.

Now, if we let $d_i^{(N)} = w + x$ and $d_{\tau(N)+1-i}^{(N)} = y + z$, where $w, x, y, z \in \mathbb{N}$, then the beginnings of a feasible assignment start to form. Based on this, we have

$$\begin{aligned} N &= d_i^{(N)} d_{\tau(N)+1-i}^{(N)} \\ &= (w + x)(y + z) \\ &= wy + xz + wz + xy \end{aligned}$$

where $i \in [\tau(N)]$. From this, it is readily seen that one possible assignment that satisfies the general constraints is

$$
\begin{aligned}
r_1 &\leftarrow wy,\\
s_0 &\leftarrow xz,\\
r_0 &\leftarrow wz, \text{ and}\\
s_1 &\leftarrow xy.
\end{aligned}
$$

Note that

$$r_1 s_0 = r_0 s_1 = wxyz$$

and that

$$
\begin{aligned}
r_1 + s_0 + r_0 + s_1 &= wy + xz + wz + xy\\
&= (w + x)(y + z)\\
&= N.
\end{aligned}
$$

Therefore, these assignments satisfy the general constraints. The concept of divisor pairs, developed earlier in this chapter, can be used to determine specific values for the variables w, x, y, and z.

5.7 Running Example: Identifying Candidate Document Collections Where $r_1 s_0 = r_0 s_1$

Earlier sections of this chapter introduced several new concepts. Subsequent sections introduce many more concepts. To aid in the comprehension of these concepts, a running example for a document collection of size 8 is used throughout the remainder of this chapter. Table 5.2 lists the divisor pair mappings for $N = 8$.

Table 5.2: Divisor Pair Mappings for $N = 8$.

$d_1^{(8)}$	$d_2^{(8)}$	$d_3^{(8)}$	$d_4^{(8)}$
$(1, 8)$	$(2, 4)$	$(4, 2)$	$(8, 1)$

We start this running example by showing how Table 5.3 on page 160 provides applications of Definition 5.5.0.5 (stated on page 153) and Definition 5.5.0.8 (stated on page 154) for selected divisor pairs. The contents of the analogous tables for Definition 5.5.0.4 (stated on page 153) and Definition 5.5.0.7 (stated on page 153), if we were to construct such tables, would be the same as those for Definitions 5.5.0.5 and 5.5.0.8, respectively, except that all 4-tuples that had at least one component with a value of 0 would not be present. The reason is because the 4-tuples in the tables that are associated with Definitions 5.5.0.5 and 5.5.0.8 can have parts with a value of 0, but the 4-tuples in the tables that are associated with Definitions 5.5.0.4 and 5.5.0.7 must have positive integers for their values.

For example, the D column for divisor pair $(2, 4)$ in the analogous table for Table 5.3, if this table existed, would only have three members:

$$(1, 1, 1, 3), (1, 1, 2, 2), (1, 1, 3, 1).$$

The reason that this example only has three members for a D column for the divisor pair $(2, 4)$ is because all four components of a D column member are required to be positive natural numbers (as contrasted with those in the table for a $\tilde{D}$ column that can have the value 0 as their lower bound). For the divisor pair $(2, 4)$, there is only one ordered pair possible for 2; that pair is 1+1. There are three ordered pairs that are possible for 4; they are 1+3, 2+2, and 3+1. These facts are evident in the list of three members above. Notice that the first 2 components of each member are ones (this corresponds to the ordered pair 1+1) and the the last 2 components correspond to the ordered pairs 1+3, 2+2, and 3+1.

5.8 Counting by the Principle of Inclusion-Exclusion

The objects to be counted are those in the union of the various sets $\tilde{G}_i^{(N)}$, where $i \in [\tau(N)]$, for N. Since the intersection of sets $\tilde{G}_i^{(N)}$ and $\tilde{G}_j^{(N)}$, where $i \neq j$ and $i, j \in [\tau(N)]$, is not necessarily disjoint, we need to use a general counting technique that computes the correct cardinality for the union of these sets, even when a particular value may be a member of more than one of the sets that are being unioned. A well-known general purpose combinatoric counting technique called the *Principle of Inclusion-Exclusion* (Riordan, 1958; Comtet, 1974; Goulden and Jackson, 1983; Stanley, 1997; Charalambides, 2002; Bóna, 2006; Aigner, 2007; Bóna, 2007) possesses this capability.

Definition 5.8.0.1. The *Principle of Inclusion–Exclusion* (POIE) is a combinatorial technique that determines the cardinality of a union of $n \in \mathbb{Z}^+$, not necessarily disjoint, sets S_1, S_2, ..., and S_n.

$$\left| \bigcup_{1 \leq i \leq n} S_i \right| = \sum_{j=1}^{n} (-1)^{j-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq n} |S_{i_1} \cap S_{i_2} \cdots \cap S_{i_j}|.$$

The notation $1 \leq i_1 < i_2 < \cdots < i_j \leq n$ means that the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of n.

From Definition 5.8.0.1, we see that, in order to use the Principle of Inclusion-Exclusion, we must be able to determine the cardinality of the intersection of the members that comprise any non-empty member of the superset of $S = \{S_1, S_2, \ldots, S_n\}$. The question arises as to what is the cardinality of 2^S, the superset of S. The answer to this question can be determined by calculating the number of distinct subsets that can be formed from a set of n distinct members.

Table 5.3: The Divisor Pairs for $N = 8$ and Their Associated $\tilde{D}$ and $\tilde{G}$ Sets.

divisor pair	$\tilde{D}$	$\tilde{G}$	divisor pair	$\tilde{D}$	$\tilde{G}$
$(1,8)$	$(0,1,0,8)$	$(0,8,0,0)$	$(8,1)$	$(0,8,0,1)$	$(0,8,0,0)$
	$(0,1,1,7)$	$(0,7,0,1)$		$(0,8,1,0)$	$(0,0,0,8)$
	$(0,1,2,6)$	$(0,6,0,2)$		$(1,7,0,1)$	$(0,7,1,0)$
	$(0,1,3,5)$	$(0,5,0,3)$		$(1,7,1,0)$	$(1,0,0,7)$
	$(0,1,4,4)$	$(0,4,0,4)$		$(2,6,0,1)$	$(0,6,2,0)$
	$(0,1,5,3)$	$(0,3,0,5)$		$(2,6,1,0)$	$(2,0,0,6)$
	$(0,1,6,2)$	$(0,2,0,6)$		$(3,5,0,1)$	$(0,5,3,0)$
	$(0,1,7,1)$	$(0,1,0,7)$		$(3,5,1,0)$	$(3,0,0,5)$
	$(0,1,8,0)$	$(0,0,0,8)$		$(4,4,0,1)$	$(0,4,4,0)$
	$(1,0,0,8)$	$(0,0,8,0)$		$(4,4,1,0)$	$(4,0,0,4)$
	$(1,0,1,7)$	$(1,0,7,0)$		$(5,3,0,1)$	$(0,3,5,0)$
	$(1,0,2,6)$	$(2,0,6,0)$		$(5,3,1,0)$	$(5,0,0,3)$
	$(1,0,3,5)$	$(3,0,5,0)$		$(6,2,0,1)$	$(0,2,6,0)$
	$(1,0,4,4)$	$(4,0,4,0)$		$(6,2,1,0)$	$(6,0,0,2)$
	$(1,0,5,3)$	$(5,0,3,0)$		$(7,1,0,1)$	$(0,1,7,0)$
	$(1,0,6,2)$	$(6,0,2,0)$		$(7,1,1,0)$	$(7,0,0,1)$
	$(1,0,7,1)$	$(7,0,1,0)$		$(8,0,0,1)$	$(0,0,8,0)$
	$(1,0,8,0)$	$(8,0,0,0)$		$(8,0,1,0)$	$(8,0,0,0)$
$(2,4)$	$(0,2,0,4)$	$(0,8,0,0)$	$(4,2)$	$(0,4,0,2)$	$(0,8,0,0)$
	$(0,2,1,3)$	$(0,6,0,2)$		$(0,4,1,1)$	$(0,4,0,4)$
	$(0,2,2,2)$	$(0,4,0,4)$		$(0,4,2,0)$	$(0,0,0,8)$
	$(0,2,3,1)$	$(0,2,0,6)$		$(1,3,0,2)$	$(0,6,2,0)$
	$(0,2,4,0)$	$(0,0,0,8)$		$(1,3,1,1)$	$(1,3,1,3)$
	$(1,1,0,4)$	$(0,4,4,0)$		$(1,3,2,0)$	$(2,0,0,6)$
	$(1,1,1,3)$	$(1,3,3,1)$		$(2,2,0,2)$	$(0,4,4,0)$
	$(1,1,2,2)$	$(2,2,2,2)$		$(2,2,1,1)$	$(2,2,2,2)$
	$(1,1,3,1)$	$(3,1,1,3)$		$(2,2,2,0)$	$(4,0,0,4)$
	$(1,1,4,0)$	$(4,0,0,4)$		$(3,1,0,2)$	$(0,2,6,0)$
	$(2,0,0,4)$	$(0,0,8,0)$		$(3,1,1,1)$	$(3,1,3,1)$
	$(2,0,3,1)$	$(2,0,6,0)$		$(3,1,2,0)$	$(6,0,0,2)$
	$(2,0,2,2)$	$(4,0,4,0)$		$(4,0,0,2)$	$(0,0,8,0)$
	$(2,0,3,1)$	$(6,0,2,0)$		$(4,0,1,1)$	$(4,0,4,0)$
	$(2,0,4,0)$	$(8,0,0,0)$		$(4,0,2,0)$	$(8,0,0,0)$

A member of S may or may not be present in one of its subsets. It is a binary decision with respect to a member's presence or absence in a subset. During subset construction, each member of S can be chosen independently of any other member of that set. The number of choices at each decision point is 2 and, thus, there are n such decisions to make. Hence, the number of ways that a subset of S can be chosen is

$$\underbrace{2 \times 2 \times \cdots \times 2}_{n} = 2^n.$$

This means that the cardinality of the superset of S is 2^n and all of its members, except one (i.e., the empty set), are nonempty sets. The cardinalities of these members range from 0 (for the empty set) to n (for the set that contains every member). Another way of deriving this identity is by noticing that the number of j-subsets in 2^S is $\binom{n}{j}$, which is the number of ways that j distinct objects, without regard to order, can be selected from n distinct ones. Therefore, the total number of ways is

$$\sum_{0 \le j \le n} \binom{n}{j} = 2^n.$$

This is a well-known identity in enumerative combinatorics (Bóna, 2007; Charalambides, 2002) and provides additional validation for the number of ways that the members of a subset of S can be chosen.

5.8.1 An Overview

This overview provides a succinct description of the reasoning behind much of the remaining part of this section. Section 5.8.6 (Entity-Relationship Models and Diagrams) is an extension of this overview. By that point, all the necessary concepts have been introduced so that the discussion there, along with the accompanying figure (i.e., Figure 5.5 on page 191) that illustrate the relationships among these concepts, are meaningful.

Each divisor pair $d_i^{(N)}$, where $i \in [\tau(N)]$, is represented by a set of 4-tuples $\tilde{D}_i^{(N)}$. The dpwcm function bijectively maps the members of $\tilde{D}_i^{(N)}$ to members of $\tilde{G}_i^{(N)}$ in such a manner that the general constraints are satisfied. The intersection of two arbitrary sets $\tilde{G}_i^{(N)}$ and $\tilde{G}_j^{(N)}$, where $i, j \in [\tau(N)]$, is the set $\tilde{G}_i^{(N)} \cap \tilde{G}_j^{(N)}$. The members of this intersection are those members of $\tilde{G}_i^{(N)}$ that are also members of $\tilde{G}_j^{(N)}$, and *vice versa.*

In order to be able to apply the POIE in a setting, it is not mandatory that the identities of the members of the sets being intersected be known. All that is required by the POIE is that there exist a way to determine the cardinality of the set that is produced by the intersection. In this section, our goal is to be able to analytically determine this cardinality, based solely on properties of the dpwcm function and the $\tilde{G}_i^{(N)}$ and $\tilde{G}_j^{(N)}$ sets. One way to accomplish this is to develop an equation, or sets of equations, that can be used to do this analytic determination. Of course, we must also prove that the determination process computes the same cardinality value, i.e., $\tilde{G}_i^{(N)} \cap \tilde{G}_j^{(N)}$, as if we intersected the actual members of $\tilde{G}_i^{(N)} \cap \tilde{G}_j^{(N)}$, and then exhaustively hand-counted how many members were in the resultant set.

5.8.2 Running Example: The Superset for a Set of Divisor Pairs and Its Cardinality

The purpose of this part of the running example is to develop some familiarity with the superset for a set of divisor pairs and the cardinality of this superset. Let

$$S = \{(1, 8), (2, 4), (4, 2), (8, 1)\}.$$

Then, 2^S (the superset of S) has $2^4 = 16$ members. The cardinalities of the members of 2^S range from 0 to 4. The superset of S has only 1 member with cardinality zero, it is

$$\{\} \text{ (the empty set).}$$

The superset has 4 members with cardinality one. They are

$$\{(1, 8)\}, \{(2, 4)\}, \{(4, 2)\}, \text{ and } \{(8, 1)\}.$$

The superset of S has 6 members with cardinality two. These members are

$\{(1, 8), (2, 4)\}, \{(1, 8), (4, 2)\}, \{(1, 8), (8, 1)\}, \{(2, 4), (4, 2)\}, \{(2, 4), (8, 1)\},$ and $\{(4, 2), (8, 1)\}.$

The superset has 4 members with cardinality three. These members are:

$$\begin{array}{ll} \{(1, 8), (2, 4), (4, 2)\}, & \{(1, 8), (2, 4), (8, 1)\}, \\ \{(1, 8), (4, 2), (8, 1)\}, & \{(2, 4), (4, 2), (8, 1)\}. \end{array}$$

The superset has 1 member with cardinality four. It is

$$\{(1, 8), (2, 4), (4, 2), (8, 1)\}.$$

In total, the superset of S has $1 + 4 + 6 + 4 + 1 = 16$ members.

5.8.3 Applicability of the Principle of Inclusion-Exclusion to This Research

The problem of determining the number of generally qualifying weak 4-compositions can be cast into a form that we can use the Principle of Inclusion-Exclusion to help solve it. This can be accomplished by recognizing that the number of divisor pairs is $\tau(N)$ and that the set that corresponds to S_{i_j}, in the context of this dissertation, is $\tilde{G}_{i_j}^{(N)}$, where $i_j \in [\tau(N)]$. The following definition is an adaptation of the POIE for the information retrieval problem that we are trying to solve.

Definition 5.8.3.1. The *Principle of Inclusion–Exclusion* for a document collection of size N, where $n = \tau(N)$, can be stated as

$$\left|\bigcup_{1\leq i\leq n} \tilde{G}_i^{(N)}\right| = \sum_{j=1}^{n}(-1)^{j-1} \sum_{1\leq i_1<i_2<\cdots<i_j\leq n} |\tilde{G}_{i_1}^{(N)} \cap \tilde{G}_{i_2}^{(N)} \cdots \cap \tilde{G}_{i_j}^{(N)}|.$$

The notation $1 \leq i_1 < i_2 < \cdots < i_j \leq n$ means that the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of n.

5.8.4 More Basic Definitions and Lemmas

Before proceeding further, we need to define the equality (and inequality) of two n-tuples and the intersection of 4-tuples. The associated definitions appear below. The use of these definitions is an integral part of the proofs of the lemmas that appear in subsequent sections of this chapter.

Definition 5.8.4.1. Let $U = (u_1, u_2, \cdots, u_n) \in \mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}$ and $V = (v_1, v_2, \cdots, v_n) \in \mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}$, where $n \in \mathbb{Z}^+$. The *equality of n-tuples* is defined as follows: two n-tuples U and V are equal if and only if $u_i = v_i$ for all $1 \leq i \leq n$; otherwise, they are not equal.

Definition 5.8.4.2. Let S_1, S_2, $\ldots$, S_n be $n \geq 1$ sets of 4-tuples. The *intersection* I of sets S_1, S_2, $\ldots$, S_n is $I = S_1 \cap S_2 \ldots \cap S_n = \{(w,x,y,z) | (w,x,y,z) \in S_1,\ (w,x,y,z) \in S_2, \ldots,$ and $(w,x,y,z) \in S_n\}$.

When sets are intersected, the result is a set that contains only those members that appear in each of the sets being intersected. Put another way, the members in the result set are those that are in common with the members of every other set in the collection of sets that are being intersected. Along these lines, we define the notions of *greatest common divisor pair* (Definition 5.8.4.3), the *set of* 4*-tuple representations of a greatest common divisor pair* (Definition 5.8.4.4 and Definition 5.8.4.5 on the following page), and the 4*-tuple generally qualifying composition of a greatest common divisor pair* (Definition 5.8.4.6 on the next page and Definition 5.8.4.7 on the following page). Basically, these notions are, respectively, multiple index extensions of these three concepts: a divisor pair, a set of 4-tuple representations of a divisor pair, and a 4-tuple generally qualifying composition of a divisor pair.

These three concepts, and their associated definitions, are described in the next several paragraphs. We start by defining a multiple index version of the greatest common divisor pair and conclude by defining multiple index versions of a divisor pair, a set of 4-tuple representations of a divisor pair, and a 4-tuple generally qualifying composition of a divisor pair.

Definition 5.8.4.3. Let $d_{i_1}^{(N)}$, $d_{i_2}^{(N)}$, $\ldots$, $d_{i_j}^{(N)}$, where $i_1, i_2, \ldots, i_j \in [\tau(N)]$, be a collection of j divisor pairs for a positive integer N. The *greatest common divisor pair*, denoted $d_{i_1,i_2,\ldots,i_j}^{(N)}$, of these divisor pairs is the 2-tuple with the value of its first component being equal to $\gcd(d_{i_1}^{(N)}[1], d_{i_2}^{(N)}[1], \ldots, d_{i_j}^{(N)}[1])$ and the value of its second component being equal to $\gcd(d_{i_1}^{(N)}[2], d_{i_2}^{(N)}[2], \ldots, d_{i_j}^{(N)}[2])$.

Definition 5.8.4.4. Let the set of 4-tuples $D_{i_1,i_2,\ldots,i_j}^{(N)}$, where $i_1, i_2, \ldots, i_j \in [\tau(N)]$, represent the set of tuples associated with greatest common divisor pair $d_{i_1,i_2,\ldots,i_j}^{(N)}$. For strong compositions, this set of 4-*tuple representations of a greatest common divisor pair* with j indices is the set

$$D^{(N)}_{i_1,i_2,\ldots,i_j} = \{(w,x,y,z) | w+x = d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \text{ and } y+z = d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \text{ and } w,x,y,z \in \mathbb{Z}^+\}.$$

Definition 5.8.4.5. Let the set of 4-tuples $\tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}$, where $i_1, i_2, \ldots, i_j \in [\tau(N)]$, represent the set of tuples associated with greatest common divisor pair $d^{(N)}_{i_1,i_2,\ldots,i_j}$. For weak compositions, this set of 4-*tuple representations of a greatest common divisor pair* with j indices is the set

$$\tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j} = \{(w,x,y,z) | w+x = d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \text{ and } y+z = d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \text{ and } w,x,y,z \in \mathbb{N}\}.$$

Definition 5.8.4.6. Let the set of 4-tuples $G^{(N)}_{i_1,i_2,\ldots,i_j}$, where $i_1, i_2, \ldots, i_j \in [\tau(N)]$, represent the set of qualifying strong 4-compositions associated with $D^{(N)}_{i_1,i_2,\ldots,i_j}$, the 4-tuple representation of a greatest common divisor pair $d^{(N)}_{i_1,i_2,\ldots,i_j}$. The 4-*tuple generally qualifying composition representation of a greatest common divisor pair* is the set

$$G^{(N)}_{i_1,i_2,\ldots,i_j} = \{\text{dpwcm}(w,x,y,z) | (w,x,y,z) \in D^{(N)}_{i_1,i_2,\ldots,i_j}\}.$$

Definition 5.8.4.7. Let the set of 4-tuples $\tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$, where $i_1, i_2, \ldots, i_j \in [\tau(N)]$, represent the set of qualifying weak 4-compositions associated with $\tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}$, the 4-tuple representation of a greatest common divisor pair $d^{(N)}_{i_1,i_2,\ldots,i_j}$. The 4-*tuple generally qualifying weak composition representation of a greatest common divisor pair* is the set

$$\tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j} = \{\text{dpwcm}(w,x,y,z) | (w,x,y,z) \in \tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}\}.$$

The second (Section 5.11.2), third (Section 5.11.3), and fourth (Section 5.11.4) cases of this subproblem rely on the notion of mutually distinct values.

Table 5.4: Sets of Divisor Pairs, Greatest Common Divisor Pairs, and Cardinalities.

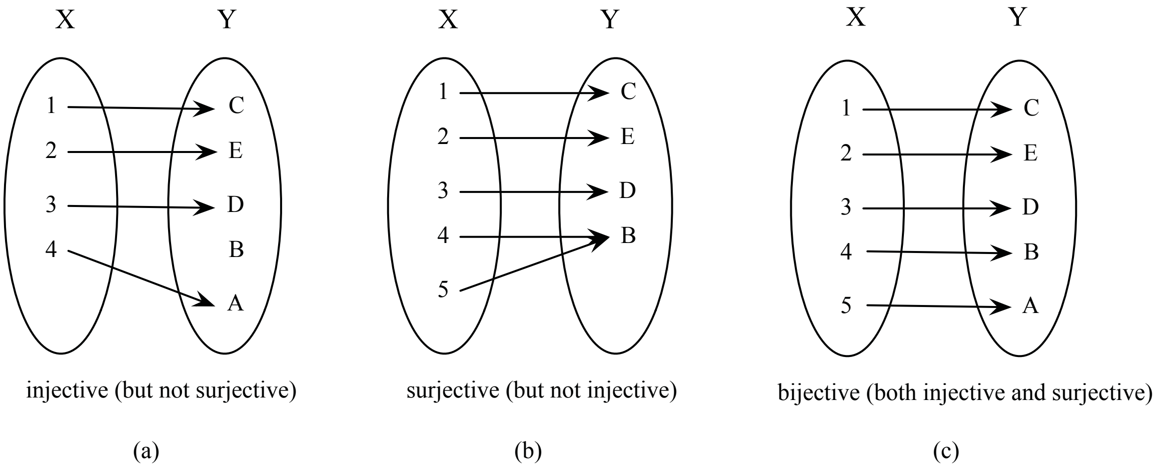
set of divisor pairs	greatest common divisor pair	number of qualifying weak compositions	number of qualifying compositions
$\{\}$	undefined	undefined	undefined
$\{(1,8)\}$	$(1,8)$	18	0
$\{(2,4)\}$	$(2,4)$	15	3
$\{(4,2)\}$	$(4,2)$	15	3
$\{(8,1)\}$	$(8,1)$	18	0
$\{(1,8),(2,4)\}$	$(1,4)$	10	0
$\{(1,8),(4,2)\}$	$(1,2)$	6	0
$\{(1,8),(8,1)\}$	$(1,1)$	4	0
$\{(2,4),(4,2)\}$	$(2,2)$	9	1
$\{(2,4),(8,1)\}$	$(2,1)$	6	0
$\{(4,2),(8,1)\}$	$(4,1)$	10	0
$\{(1,8),(2,4),(4,2)\}$	$(1,2)$	6	0
$\{(1,8),(2,4),(8,1)\}$	$(1,1)$	4	0
$\{(1,8),(4,2),(8,1)\}$	$(1,1)$	4	0
$\{(2,4),(4,2),(8,1)\}$	$(2,1)$	6	0
$\{(1,8),(2,4),(4,2),(8,1)\}$	$(1,1)$	4	0

Definition 5.8.4.8. Let the values in $V = \{v_1, v_2, \ldots, v_m | m \in \mathbb{Z}^+\}$ be called *mutually distinct* if and only if $|V| = m$. That is, $v_i = v_j$ if and only if $i = j$ where $1 \leq i, j \leq m$.

Definition 5.8.4.9. Let $f : X \rightarrow Y$ be a function f from a set X to a set Y. Then f is an *injective* function, or *injection*, with the property that, for every $y \in Y$, there is at most one $x \in X$ such that $f(x) = y$.

Definition 5.8.4.10. Let $f : X \rightarrow Y$ be a function f from a set X to a set Y. Then f is an *surjective* function, or *surjection*, with the property that, for every $y \in Y$, there is at least one $x \in X$ such that $f(x) = y$.

Definition 5.8.4.11. Let $f : X \rightarrow Y$ be a function f from a set X to a set Y. Then f is an *bijective* function, or *bijection*, with the property that, for every $y \in Y$, there is exactly one $x \in X$ such that $f(x) = y$.



injective (but not surjective)

(a)

surjective (but not injective)

(b)

bijective (both injective and surjective)

(c)

Figure 5.1: Injective, Surjective, and Bijective Functions.

For some input values, dpwcm, the divisor pair weak composition mapping function, when applied to those values, yields the same result when those inputs are scaled in certain ways. This fact is important in several of the proofs to follow because it allows the rewriting of dpwcm expressions in some instances. The associated lemmas and their proofs are as follows.

The next two lemmas (i.e., Lemma 5.8.1 and Lemma 5.8.2 on page 171) enable the rewriting of some weak 4-compositions by proving that, under certain circumstances, different ways of expressing these weak compositions are equivalent. These results are used in subsequent parts of this chapter to help prove other lemmas.

Lemma 5.8.1. *Suppose* $a, b, c, d \in \mathbb{N}$; $m \in \mathbb{Z}^+$; *and* m *is a positive divisor of* $\gcd(c, d)$. *Then* $\text{dpwcm}(ma, mb,\ c/m, d/m) = \text{dpwcm}(a, b, c, d)$.

Proof. By Definition 5.5.0.6 on page 153, we can write

$$\begin{aligned}
\text{dpwcm}(ma, mb, c/m, d/m) &= ((ma)(c/m), (mb)(d/m), (ma)(d/m), (mb)(c/m)) \\
&= ((ac)(m/m), (bd)(m/m), (ad)(m/m), (bc)(m/m)) \\
&= (ac, bd, ad, bc). \\
&= \text{dpwcm}(a, b, c, d).
\end{aligned}$$

Since $\gcd(c, d)$ denotes the greatest common denominator of c and d, then any positive divisor m of this gcd also evenly divides c and d. □

Example Illustrating How Three Weak 4-Compositions Can Be Equivalent Under the dpwcm Mapping When $N = 12$

Assume that $N = 12$ and that the 4-tuple $(0, 1, 4, 8)$ is one of the weak 4-compositions that is associated with N. We find below that $\text{dpwcm}(0, 1, 4, 8)$ yields $(0, 8, 0, 4)$; that is,

$$\begin{aligned}
\text{dpwcm}(0, 1, 4, 8) &= (0 \cdot 4, 1 \cdot 8, 0 \cdot 8, 1 \cdot 4) \\
&= (0, 8, 0, 4) \qquad (5.8.1)
\end{aligned}$$

by Definition 5.5.0.6 on page 153.

We also find that applying the dpwcm function to the weak 4-compositions $(0, 2, 2, 4)$ and $(0, 4, 1, 2)$ yields the 4-tuple $(0, 8, 0, 4)$ in each instance, the very same result that it yielded when it was applied earlier to the weak 4-composition $(0, 1, 4, 8)$; that is,

$$\begin{aligned} \operatorname{dpwcm}(0,1,4,8) &= \operatorname{dpwcm}(2\cdot 0, 2\cdot 1, 4/2, 8/2) \\ &= \operatorname{dpwcm}(0,2,2,4) \\ &= (0\cdot 2, 2\cdot 4, 0\cdot 4, 2\cdot 2) \\ &= (0,8,0,4) \end{aligned} \tag{5.8.2}$$

and

$$\begin{aligned} \operatorname{dpwcm}(0,1,4,8) &= \operatorname{dpwcm}(4\cdot 0, 4\cdot 1, 4/4, 8/4) \\ &= \operatorname{dpwcm}(0,4,1,2) \\ &= (0\cdot 1, 4\cdot 2, 0\cdot 2, 4\cdot 1) \\ &= (0,8,0,4). \end{aligned} \tag{5.8.3}$$

The reason that the same value was yielded with each application of the dpwcm function is because the last two values in $(0, 1, 4, 8)$ are the integers 4 and 8. The greatest common divisor of these two values is 4 (i.e., $\gcd(4, 8) = 4$). The positive integer divisors of 4 are the numbers 1, 2, and 4.

By Lemma 5.8.1 on the previous page, Equation 5.8.1 on the preceding page can be rewritten as Equation 5.8.2 with the scaling factor m having the value 2. In essence, this means that

$$\operatorname{dpwcm}(0, 1, 4, 8) = \operatorname{dpwcm}(0, 2, 2, 4).$$

Similarly, this lemma can be used to rewrite Equation 5.8.1 as Equation 5.8.3. In this case, the scaling factor m is 4. In essence, this means that

$$\mathrm{dpwcm}(0,1,4,8) = \mathrm{dpwcm}(0,4,1,2).$$

Lemma 5.8.2. *Suppose $a, b, c, d \in \mathbb{N}$; $m \in \mathbb{Z}^+$; and m is a positive divisor of* $\gcd(a, b)$. *Then* $\mathrm{dpwcm}(a/m,\ b/m,\ mc, md) = \mathrm{dpwcm}(a, b, c, d)$.

Proof. By Definition 5.5.0.6 on page 153, we can write

$$\begin{aligned}\mathrm{dpwcm}(a/m, b/m, mc, md) &= ((a/m)(mc), (b/m)(md), (a/m)(md), (b/m)(mc)) \\ &= ((ac)(m/m), (bd)(m/m), (ad)(m/m), (bc)(m/m)) \\ &= (ac, bd, ad, bc) \\ &= \mathrm{dpwcm}(a, b, c, d).\end{aligned}$$

Since $\gcd(a, b)$ denotes the greatest common denominator of a and b, then any positive divisor m of this gcd also divides a and b. □

Example Illustrating How Two Weak 4-Compositions Can Be Equivalent Under the dpwcm Mapping When $N = 12$

Assume that $N = 12$ and that the 4-tuple $(2, 4, 1, 1)$ is one of the weak 4-compositions that is associated with that number. We find below that $\mathrm{dpwcm}(2, 4, 1, 1)$ yields $(2, 4, 2, 4)$; that is,

$$\begin{aligned}\mathrm{dpwcm}(2,4,1,1) &= (2\cdot 1, 4\cdot 1, 2\cdot 1, 4\cdot 1) \\ &= (2,4,2,4), \qquad (5.8.4)\end{aligned}$$

by Definition 5.5.0.6 on page 153.

We also find that applying the dpwcm function to the weak 4-composition $(1, 2, 2, 2)$

yields the 4-tuple $(2, 4, 2, 4)$, the very same result that it yielded when it was applied earlier to the weak 4-composition $(2, 4, 1, 1)$; that is,

$$\begin{aligned}\operatorname{dpwcm}(2,4,1,1) &= \operatorname{dpwcm}(2/2, 4/2, 2\cdot 1, 2\cdot 1)\\ &= \operatorname{dpwcm}(1,2,2,2)\\ &= (1\cdot 2, 2\cdot 2, 1\cdot 2, 2\cdot 2)\\ &= (2,4,2,4). \qquad (5.8.5)\end{aligned}$$

The reason that the same value was yielded with each application of the dpwcm function is because the first two values in $(2, 4, 1, 1)$ are the integers 2 and 4. The greatest common divisor of these two values is 2 (i.e., $\gcd(2, 4) = 2$). The positive integer divisors of 2 are the numbers 1 and 2.

By Lemma 5.8.2 on the preceding page, Equation 5.8.4 on the previous page can be rewritten as Equation 5.8.5 with the scaling factor m having the value 2. In essence, this means that

$$\operatorname{dpwcm}(2,4,1,1) = \operatorname{dpwcm}(1,2,2,2).$$

5.8.5 Lemmas for the Establishment of Essential Bijections

The next lemma establishes a bijection between the 4-tuple representation $\tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}$ of the greatest common divisor pair for the divisor pairs that are associated with the values of the j indices i_1, i_2, ..., i_j and the set of 4-tuple generally qualifying weak composition representation $\tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$ of the greatest common divisor pair for these indices.

Lemma 5.8.3. *Suppose $r = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of r (i.e., $1 \le i_1 < i_2 < \cdots < i_j \le r$). Then the function*

$$\operatorname{dpwcm} : \tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j} \to \tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$$

is bijective.

Proof. A bijective function is one that is both surjective and injective. First, we prove that the dpwcm function is surjective. After that, we prove that it is also injective.

By Definition 5.8.4.7 on page 166, the function dpwcm maps each member of set $\tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}$ to at least one of the members in set $\tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$. A member in set $\tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$ can exist only if it is mapped to by a member of set $\tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}$. Therefore, the function

$$\text{dpwcm} : \tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j} \rightarrow \tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$$

is surjective.

We use *proof by contradiction* for the injective part of this result. Assume that the dpwcm function is not injective. Then there exists 4-tuples $g_1, g_2 \in \tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}$ and $h \in \tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$, with g_1 not equal to g_2, such that $\text{dpwcm}(g_1[1], g_1[2], g_1[3], g_1[4]) = h$ and $\text{dpwcm}(g_2[1], g_2[2], g_2[3], g_2[4]) = h$. This assumption means that there exists at least one value for i in the set $\{1, 2, 3, 4\}$ such that the value of $g_1[i]$ is different than the value for $g_2[i]$. Since each component in g_1 can be the same, or different, than its counterpart in g_2, and there are four of these components, then there are $2^4 = 16$ possible events. Table 5.5 on the following page enumerates these events. Possibility 16 cannot be a candidate because the corresponding components are all in agreement.

Most of these other 15 events cannot occur, though, due to the general requirements that

$$g_1[1] + g_1[2] = g_2[1] + g_2[2] = d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$$

and that

$$g_1[3] + g_1[4] = g_2[3] + g_2[4] = d^{(N)}_{i_1,i_2,\ldots,i_j}[2].$$

If $g_1[1]$ has the same value as $g_2[1]$, then $g_1[2]$ must have the same value as $g_2[2]$. Likewise, if $g_1[1]$ has a different value than $g_2[1]$, then $g_1[2]$ must have a different value than $g_2[2]$.

Similar relations hold for the sums $g_1[3] + g_1[4]$ and $g_2[3] + g_2[4]$.

An inspection of Table 5.5 reveals that Possibilities 2, 3, 5-12, 14, and 15 cannot occur because of the general requirements that were just enumerated in the immediately prior paragraph. As was mentioned earlier, Possibility 16 can be eliminated because it represents the situation where g_1 and g_2 are equal; our assumption that dpwcm is not injective implies that g_1 and g_2 cannot be equal. This leaves only three events to explore. The analysis for each of them appears as a separate case below.

Table 5.5: List of the Sixteen Possibilities for Matches/Differences between the Values of the Corresponding Components (N = no, *blank*=yes).

possibility	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
$g_1[1] = g_2[1]$*?*	N	N	N	N	N	N	N	N								
$g_1[2] = g_2[2]$*?*	N	N	N	N					N	N	N	N				
$g_1[3] = g_2[3]$*?*	N	N			N	N			N	N			N	N		
$g_1[4] = g_2[4]$*?*	N		N		N		N		N		N		N		N	

Analysis for when none of the values of the corresponding components of g_1 and g_2 are equal. This is Possibility 1.

Included among the requirements for this case is the requirement that no pair of corresponding components of g_1 and g_2 can be equal (i.e., $g_1[1] \neq g_2[1]$, $g_1[2] \neq g_2[2]$, and so on). By Lemma 5.8.1 on page 169,

$$\text{dpwcm}(g_1[1], g_1[2], g_1[3], g_1[4])$$

can always be rewritten as

$$\text{dpwcm}(m_1 g_1[1], m_1 g_1[2], g_1[3]/m_1, g_1[4]/m_1),$$

where m_1 represents one of possibly many divisors of the value that is represented by the greatest common denominator of $g_1[3]$ and $g_1[4]$.

Similarly, by Lemma 5.8.2 on page 171,

$$\mathrm{dpwcm}(g_1[1], g_1[2], g_1[3], g_1[4])$$

can also be rewritten as

$$\mathrm{dpwcm}(g_1[1]/m_2, g_1[2]/m_2, m_2 g_1[3], m_2 g_1[4]),$$

where m_2 represents one of possibly many divisors of the value that is represented by the greatest common denominator of $g_1[1]$ and $g_1[2]$.

No matter whether Lemma 5.8.1 on page 169 or Lemma 5.8.2 on page 171 is used to rewrite

$$\mathrm{dpwcm}(g_1[1], g_1[2], g_1[3], g_1[4]),$$

the respective m-value must be greater than 1 because both

$$(m_1 g_1[1], m_1 g_1[2], g_1[3]/m_1, g_1[4]/m_1),$$

and

$$(g_1[1]/m_2, g_1[2]/m_2, m_2 g_1[3], m_2 g_1[4]),$$

are required to be different than

$$(g_1[1], g_1[2], g_1[3], g_1[4]).$$

This means that

$$\begin{aligned}(g_2[1], g_2[2], g_2[3], g_2[4]) &= (m_1 g_1[1], m_1 g_1[2], g_1[3]/m_1, g_1[4]/m_1) \\ &= h,\end{aligned}$$

or that

$$\begin{aligned}(g_2[1], g_2[2], g_2[3], g_2[4]) &= (g_1[1]/m_2, g_1[2]/m_2, m_2 g_1[3], m_2 g_1[4]) \\ &= h.\end{aligned}$$

The other requirement is that, in any rewrite, the sums of the first two components must equal $d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ and the sums of the last two components must equal $d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$. This can only occur when the m-value is 1. If the value of m_1 is greater than 1, then the sum $m_1 g_1[1] + m_1 g_1[2]$ is greater than $d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$. Similarly, if the value of m_2 is greater than 1, then the sum $m_2 g_1[3] + m_2 g_1[4]$ is greater than $d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$. Hence, the assumption that g_1 and g_2 map to 4-tuples that are equal is false.

Analysis for when the values of the first two corresponding components of g_1 and g_2 are not equal but the values for each of the remaining two are equal. This is Possibility 4.

This possibility means that g_1 equals

$$(v_1, d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - v_1, v_3, v_4)$$

and that g_2 equals

$$(w_1, d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - w_1, v_3, v_4),$$

where $v_1 \neq w_1$, $v_3 + v_4 = d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$. The variables v_1, v_3, v_4, and w_1 are members of $\mathbb{N}$.

By Definition 5.5.0.6 on page 153, dpwcm, the generally qualifying composition mapping function, maps g_1 to

$$(v_1v_3, (d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - v_1)v_4, v_1v_4, (d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - v_1)v_3).$$

It also maps g_2 to

$$(w_1v_3, (d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - w_1)v_4, w_1v_4, (d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - w_1)v_3).$$

In order to complete this part of the proof, we need to investigate the value of g_1 and g_2 for these three cases: (1) $v_3 = 0$ and $v_4 \neq 0$; (2) $v_3 \neq 0$ and $v_4 = 0$; and (3) $v_3 \neq 0$ and $v_4 \neq 0$. Note that, because the sum $v_3 + v_4$ must be a positive integer, at least one of v_3 and v_4 must have a value that is greater than 0. The values of v_3 and v_4 in Case 1 imply that $v_1v_4 \neq w_1v_4$; the values in Case 2 imply that $v_1v_3 \neq w_1v_3$; and the values in Case 3 imply that $v_1v_4 \neq w_1v_4$ and $v_1v_3 \neq w_1v_3$. Collectively, each of these cases means that there is at least one component in g_1 that has a different value than its counterpart in g_2. Hence, the assumption that g_1 and g_2 map to 4-tuples that are equal is false.

Analysis for when the values of the first two corresponding components of g_1 and g_2 are equal but the values for each of the remaining two are not equal. This is Possibility 13.

This means that g_1 equals

$$(v_1, v_2, v_3, d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - v_3)$$

and that g_2 equals

$$(v_1, v_2, w_3, d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - w_3)$$

where $v_1 + v_2 = d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ and $v_3 \neq w_3$. The variables v_1, v_2, v_3, and w_3 are members of $\mathbb{N}$.

By Definition 5.5.0.6 on page 153, dpwcm, the generally qualifying composition mapping function, maps g_1 to

$$(v_1 v_3, v_2(d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - v_3), v_1(d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - v_3), v_2 v_3).$$

It also maps g_2 to

$$(v_1 w_3, v_2(d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - w_3), v_1(d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - w_3), v_2 w_3).$$

In order to complete this part of the proof, we need to investigate the value of g_1 and g_2 for these three cases: (1) $v_1 = 0$ and $v_2 \neq 0$; (2) $v_1 \neq 0$ and $v_2 = 0$; and (3) $v_1 \neq 0$ and $v_2 \neq 0$. Note that, because the sum $v_1 + v_2$ must be a positive integer, at least one of v_1 and v_2 must have a value that is greater than 0. The values of v_1 and v_2 in Case 1 imply that $v_2 v_3 \neq v_2 w_3$; the values in Case 2 imply that $v_1 v_3 \neq v_1 w_3$; and the values in Case 3 imply that $v_2 v_3 \neq v_2 w_3$ and $v_1 v_3 \neq v_1 w_3$. Collectively, each of these cases means that there is at least one component in g_1 that has a different value than its counterpart in g_2. Hence, the assumption that g_1 and g_2 map to 4-tuples that are equal is false.

Summary.

The above cases that are associated with Possibilities 1, 4, and 13 show that the assumption that the function

$$\text{dpwcm} : \tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j} \to \tilde{G}^{(N)}_{i_1,i_2,\ldots,i_j}$$

is not injective leads to various contradictions. Hence, the function dpwcm must be injective. Now that we have shown that the function is both injective and surjective, we can conclude that it is bijective. □

Note that Figure 5.2 is an example of one of the many ways that the bijection between sets such as $\tilde{D}_2^{(8)}$ and $\tilde{G}_2^{(8)}$ can be depicted. This relationship was established by Lemma 5.8.3 on page 172. The next two lemmas also establish bijective relationships. The relationships established by Lemma 5.8.3 on page 172 and the next two lemmas (i.e., Lemma 5.8.4 on the next page and Lemma 5.8.5 on page 181) are crucial in the development of some of the counting expressions that appear later in this chapter.

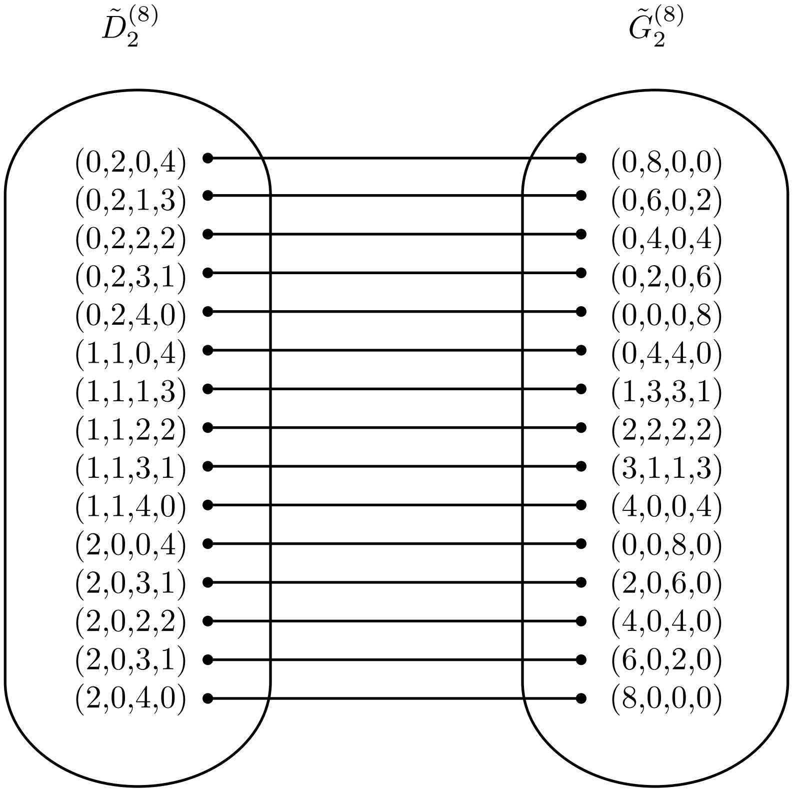


Figure 5.2: The Bijection Between Sets $\tilde{D}_2^{(8)}$ and $\tilde{G}_2^{(8)}$. The dpwcm function maps each member of the former set to its corresponding member in the latter set.

Lemma 5.8.4 on the next page, the next lemma, shows that there is a bijection between any set of weak 4-compositions and another set of weak 4-compositions, if this latter set is constructed in a certain way. For each weak 4-composition c in the original set, a weak 4-composition is created for an initially empty new set. This member of the new set is a weak 4-composition where the values of the first two components are the respective values of the first two components of c, except that they have both been scaled by the same arbitrary positive real number a. Similarly, the last two components of the new weak 4-composition have, as their respective values, the scaled values of the last two components of the original weak 4-composition c. The scaling factor in this case is also a positive real number and is denoted as b. The lemma also establishes that this transformation is valid whether the values of a and b are the same, or different. The main use of this lemma is to help with the proof of Lemma 5.8.5 on the following page.

Lemma 5.8.4. *Suppose X is a non-empty set of 4-compositions, variables a and b are positive real numbers, and*

$$Y = \{f(w, x, y, z) | (w, x, y, z) \in X \textit{ and } f(w, x, y, z) = (aw, ax, by, bz)\}.$$

Then there is a bijection between sets X and Y.

Proof. By the definition of X and Y, the function f is surjective because it maps at least one member of set X to each member of set Y. The second part of this proof establishes that the mapping induced by function f is also injective. The technique used is proof by contradiction.

Let (w_1, x_1, y_1, z_1) and (w_2, x_2, y_2, z_2) both be members of X. Also, let $a \in \mathbb{R}^+$ and $b \in \mathbb{R}^+$ be the scaling factors for the variables w_1, x_1, w_2, x_2 and the variables y_1, z_1, y_2, z_2, respectively. Assume that

$$(w_1, x_1, y_1, z_1) \neq (w_2, x_2, y_2, z_2),$$

but that

$$f(w_1, x_1, y_1, z_1) = f(w_2, x_2, y_2, z_2).$$

Since the scaling factor value for the first two components is independent of the scaling factor value for the last 2 components, the analyses for these two groups of components can be handled separately.

Assume that $(w_1, x_1) \neq (w_2, x_2)$ but that $(aw_1, ax_1) = (aw_2, ax_2)$. This implies that $aw_1 = aw_2$ and $ax_1 = ax_2$. Due to a being a positive value, it can be further stated that $w_1 = w_2$ and $x_1 = x_2$ must also hold. However, this contradicts the assumption that was made at the beginning of this paragraph because, if both of these conditions hold, then $(w_1, x_1) \neq (w_2, x_2)$ must be false.

Now, assume that $(y_1, z_1) \neq (y_2, z_2)$ but that $(by_1, bz_1) = (by_2, bz_2)$. This implies that $by_1 = by_2$ and $bz_1 = bz_2$. Due to b being a positive value, it can be further stated that $w_1 = w_2$ and $z_1 = z_2$ must also hold. But, this contradicts the assumption that was made at the beginning of this paragraph because if both of these conditions hold, then the assertion $(w_1, x_1) \neq (w_2, x_2)$ must be false.

From the two cases above, it has been established that every member of X maps to at least one member of Y and, furthermore, that no two members of Y map to the same member of X (i.e., the mapping is injective). Hence, the mapping between X and Y is bijective. □

The lemma below establishes that, for a non-empty subset of divisor pairs, identified by j indices, a bijection exists between the intersection of the set of generally qualifying weak 4-compositions associated with the divisor pairs and the set of 4-tuple representations of the the greatest common divisor pair for the divisor pairs that are associated with these j indices.

Lemma 5.8.5. *Suppose $r = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of r (i.e., $1 \leq i_1 < i_2 < \cdots < i_j \leq r$). Then there exists a bijection between the set $\tilde{G}_{i_1}^{(N)} \cap \tilde{G}_{i_2}^{(N)} \cap \cdots \cap \tilde{G}_{i_j}^{(N)}$ and the set $\tilde{D}_{i_1,i_2,\ldots,i_j}^{(N)}$. The cardinality of $\tilde{D}_{i_1,i_2,\ldots,i_j}^{(N)}$ is $(c_1 + 1)(c_2 + 1)$ where $c_1 = d_{i_1,i_2,\ldots,i_j}^{(N)}[1]$ and $c_2 = d_{i_1,i_2,\ldots,i_j}^{(N)}[2]$).*

Proof. Definition 5.5.0.5 on page 153, Definition 5.5.0.6 on page 153, and Definition 5.5.0.8 on page 154 establish the relationship between the 4-tuple representation $\tilde{D}_i^{(N)}$ of a divisor pair $d_i^{(N)}$ and its corresponding weak 4-composition representation $\tilde{G}_i^{(N)}$. Let the notation

$$\tilde{I} = \tilde{I}_{i_1,i_2,\ldots,i_j}^{(N)} = \tilde{G}_{i_1}^{(N)} \cap \tilde{G}_{i_2}^{(N)} \cap \cdots \cap \tilde{G}_{i_j}^{(N)}$$

denote a set that contains only the weak 4-compositions that are a member of every one of the sets being intersected. In other words, $\tilde{I}_{i_1,i_2,\ldots,i_j}^{(N)}$ contains only the weak 4-compositions that are common to all of the $\tilde{G}_{i_j}^{(N)}$ where $j \in [r]$. The shorthand $\tilde{I}$ is used for the notation $\tilde{I}_{i_1,i_2,\ldots,i_j}^{(N)}$ unless there is a statement to the contrary..

The corresponding collection of j divisor pairs $d_{i_j}^{(N)}$, where $j \in [r]$, can be represented, one per line, is composed of these pairs.

$$(d_{i_1}^{(N)}[1]\ ,\ d_{i_1}^{(N)}[2]) \tag{5.8.6}$$

$$(d_{i_2}^{(N)}[1]\ ,\ d_{i_2}^{(N)}[2]) \tag{5.8.7}$$

$$\cdots \tag{5.8.8}$$

$$(d_{i_j}^{(N)}[1]\ ,\ d_{i_j}^{(N)}[2]). \tag{5.8.9}$$

The "set intersection" analog for the divisor pairs listed just above is the "greatest common divisor operation." It is used to compute the largest common factor over the first component of the respective divisor pairs and the largest common factor over the second component of the respective divisor pairs. The expression

$$c_1 = c_{1,\{i_1,i_2,\ldots,i_j\}} = \gcd(d_{i_1}^{(N)}[1], d_{i_2}^{(N)}[1], \ldots, d_{i_j}^{(N)}[1]) \tag{5.8.10}$$

computes the largest common factor over the first components and the expression

$$c_2 = c_{2,\{i_1,i_2,\ldots,i_j\}} = \gcd(d_{i_1}^{(N)}[2], d_{i_2}^{(N)}[2], \ldots, d_{i_j}^{(N)}[2]) \tag{5.8.11}$$

does the same over the second components. The shorthand c_1 and c_2 are used for $c_{1,\{i_1,i_2,\ldots,i_j\}}$ and $c_{2,\{i_1,i_2,\ldots,i_j\}}$, respectively, unless there is a statement to the contrary.

Divisor pairs 5.8.6 to 5.8.9 on the preceding page, by the use of these largest common factors, can be rewritten as

$$\begin{gathered}
((d_{i_1}^{(N)}[1]/c_1)\, c_1, (d_{i_1}^{(N)}[2]/c_2)\, c_2) \\
((d_{i_2}^{(N)}[1]/c_1)\, c_1, (d_{i_2}^{(N)}[2]/c_2)\, c_2) \\
\cdots \\
((d_{i_j}^{(N)}[1]/c_1)\, c_1, (d_{i_j}^{(N)}[2]/c_2)\, c_2),
\end{gathered}$$

respectively. Note that the values for c_1 and c_2 are equal to the values of the first and second components, respectively, of the greatest common divisor pair $d_{i_1,i_2,\ldots,i_j}^{(N)}$ for this collection of j divisor pairs, that is,

$$c_1 = \gcd(d_{i_1}^{(N)}[1], d_{i_2}^{(N)}[1], \ldots, d_{i_j}^{(N)}[1]) = d_{i_1,i_2,\ldots,i_j}^{(N)}[1]$$

and

$$c_2 = \gcd(d_{i_1}^{(N)}[2], d_{i_2}^{(N)}[2], \ldots, d_{i_j}^{(N)}[2]) = d_{i_1,i_2,\ldots,i_j}^{(N)}[2].$$

Also, note that the $d_{i_j}^{(N)}[1]/c_1$ and $d_{i_j}^{(N)}[2]/c_2$ values, where $i_j \in [\tau(N)]$, are positive integers.

From this rewrite, and by Equation 2.2.2 on page 26, the cardinality of $\tilde{I}$ is

$$|\tilde{I}| = \binom{c_1 + 2 - 1}{2 - 1}\binom{c_2 + 2 - 1}{2 - 1} = \binom{c_1 + 1}{2 - 1}\binom{c_2 + 1}{2 - 1} = (c_1 + 1)(c_2 + 1)$$

because there are c_1+1 possible weak 2-compositions for $d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$, c_2+1 possible weak 2-compositions for $d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$, and these sets of weak compositions are independent.

Now, let

$$\tilde{I}' = \tilde{I}^{(N)\prime}_{i_1,i_2,\ldots,i_j} = \{(w,x,y,z) \in \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N} | w+x = c_1 \text{ and } y+z = c_2\}.$$

By Definition 5.8.4.5 on page 166, this is equivalent to writing $\tilde{I}^{(N)\prime}_{i_1,i_2,\ldots,i_j} = \tilde{D}^{(N)}_{i_1,i_2,\ldots,i_j}$. The shorthand $\tilde{I}'$ is used for the notation $\tilde{I}^{(N)\prime}_{i_1,i_2,\ldots,i_j}$ when it is clear from the context of use that they represent the same concept. Each divisor pair $d^{(N)}_{i_j}$, in a collection of $j \in [m]$ divisor pairs $\{d^{(N)}_{i_1}, d^{(N)}_{i_2}, \ldots, d^{(N)}_{i_j}\}$, and using a 4-tuple representation, has the set of mapping functions

$$L^{(N)}_{i_j} = \{f_{i_j}(w,x,y,z) | (w,x,y,z) \in \tilde{I}'\}$$

associated with it where the function f_{i_j} is defined as

$$f_{i_j}(w,x,y,z) \rightarrow ((d^{(N)}_{i_j}[1]/c_1)\,w, (d^{(N)}_{i_j}[1]/c_1)\,x, (d^{(N)}_{i_j}[2]/c_2)\,y, (d^{(N)}_{i_j}[2]/c_2)\,z).$$

By Lemma 5.8.4 on page 180, the mapping between members of $\tilde{I}'$ and $L^{(N)}_{i_j}$ is bijective.

There is also a bijective relationship between sets $\tilde{I}$ and $\tilde{I}'$. By Equation 2.2.2 on page 26, the cardinality of $\tilde{I}'$ is the same as that of $\tilde{I}$, that is,

$$|I'| = |\tilde{I}| = \binom{c_1+2-1}{2-1}\binom{c_2+2-1}{2-1} = \binom{c_1+1}{2-1}\binom{c_2+1}{2-1} = (c_1+1)(c_2+1).$$

Also, by that same equation, the cardinality of $L^{(N)}_{i_j}$ is

$$|L^{(N)}_{i_j}| = \binom{c_1+1}{2-1}\binom{c_2+1}{2-1} = (c_1+1)(c_2+1).$$

Of course, it is no coincidence that the cardinalities of both $\tilde{I}'$ and $L_{i_j}^{(N)}$ are the same because prior discussions in the proof for this lemma have established that the function f_{i_j} is bijective.

Assume that $L_{i_j}^{(N)}$ and $\tilde{I}'$ exist; that $j \in [m]$; and that (w, x, y, z) is an arbitrary member of $\tilde{I}'$. The expression $M_{i_j}^{(N)}$ is used to denote the members of $\tilde{G}_{i_j}^{(N)}$ that are also members of $\tilde{I}$.

$$\begin{aligned} M_{i_j}^{(N)} &= \{(\text{dpwcm}(a, b, c, d) | (a, b, c, d) \in L_{i_j}^{(N)}\} \\ &= \{(ac, bd, ad, bc) | (a, b, c, d) \in L_{i_j}^{(N)}\} \\ &= \tilde{I}. \end{aligned}$$

The above equation for $M_{i_j}^{(N)}$ states that if (a, b, c, d) is an arbitrary member of $L_{i_j}^{(N)}$, then the $M_{i_j}^{(N)}$ member that the dpwcm function maps it to is (ac, bd, ad, bc). Furthermore, $M_{i_j}^{(N)}$ and $\tilde{I}$ are identical sets. Moreover,

$$M_{i_1}^{(N)} = M_{i_2}^{(N)} = \cdots = M_{i_j}^{(N)} = \tilde{I}$$

because the 4-tuples that are in $\tilde{I}$ are exactly those 4-tuples that are in the set

$$\tilde{G}_{i_1}^{(N)} \cap \tilde{G}_{i_2}^{(N)} \cap \cdots \cap \tilde{G}_{i_j}^{(N)}.$$

□

5.8.6 Entity-Relationship Models and Diagrams

There are many concepts that are introduced in the remainder of this section. The entity-relationship model (ERM) (Chen, 1976) is used to model some of the semantic

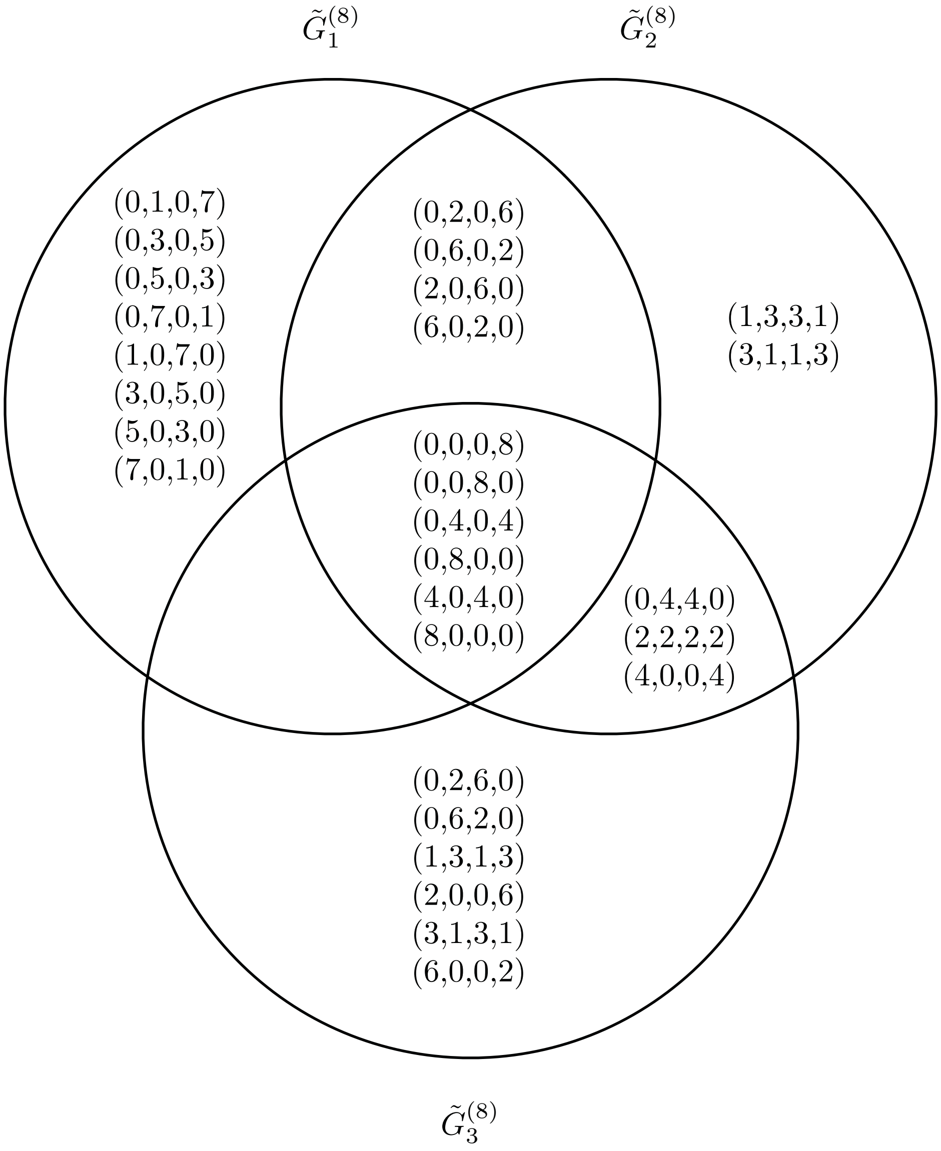


Figure 5.3: Example of the Intersection Between Three $\tilde{G}$ Sets When $N = 8$.

relationships between them. Historically, the ERM has been mainly used in the relational database community to model relationships between entities in a database.

An ERM is realized by an entity-relationship (ER) diagram. There are many notations to represent ER diagrams. Some of the more widely used ones are Chen notation (Chen, 1976), IDEFIX notation (Bruce, 1992), Bachman notation (Bachman, 1969), Martin notation (Martin, 1990), (min,max)-notation (Batini et al., 1992; McFadden and Hoffer, 1994; Teorey, 1991), the notation used in the UML standard (Jacobson et al., 1999), and EXPRESS notation (Schenck and Wilson, 1994). Common among these different notations are that rectangles represent entities. Where these notations mainly differ is in how they represent relationships between entities (Song et al., 1995).

The notation used in an ER diagram is often not sufficient to explain all that is necessary about the relationships between its entities. Typically, the notation suffices to explain most aspects of these relationships. What cannot be sufficiently detailed is normally explained in accompanying documentation.

Definition 5.8.6.1. A *one-to-one relationship (1:1)* from entity type X to entity type Y is one in which an X entity maps to at most one Y entity and *vice versa.*

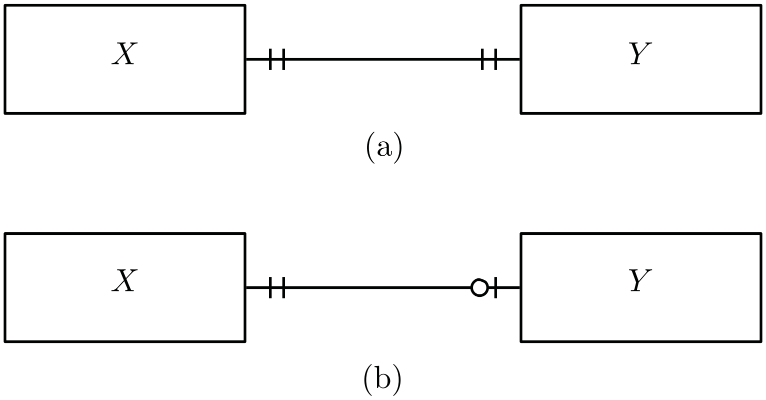
Definition 5.8.6.2. A *one-to-many relationship (1:m)* from entity type X to entity type Y is one in which an X entity can map to any number of Y entities (including zero) and any Y entity can map to at most one X entity.

Definition 5.8.6.3. A *many-to-one relationship (m:1)* from entity type X to entity type Y is one in which an X entity can map to at most one Y entity but a Y entity can map to any number of X entities.

Definition 5.8.6.4. A *many-to-many relationship (m:n)* from entity type X to entity type Y is one in which an X entity can map to any number (including zero) of Y entities and *vice versa.*

The entity-relationship (ER) diagram in Figure 5.5 on page 191 depicts many important concepts from Definitions 5.5.0.1 to 5.5.0.8 on pages 152–154, inclusive, and how they are related. These concepts may be somewhat abstract to the reader. Before proceeding farther, it would be helpful to discuss the notation in the figure and to provide an example to illustrate various aspects of these concepts. The rectangles in this diagram represent entities, the diamonds represent relationships, and the labels on the connecting lines represent ordinality and cardinality constraints.

Cardinality refers to the number of instances of one entity type that relate to one instance of another entity type, whereas *ordinality* refers to whether the relationship is optional or mandatory (White, 1994). Both of these terms deal with the number of occurrences of a relationship. The ordinality value can be viewed as specifying the minimum number of relationships, the cardinality value can be viewed as specifying the maximum number of relationships. If the ordinality value is allowed to be 0, then the relationship is optional. But, if the value is one or greater, the relationship is mandatory. Figure 5.4 describes notation that is used later in Figure 5.5.



(a)

(b)

Figure 5.4: ER Notation. Figure 5.4(a) states that each entity in sets X and Y is related to exactly one entity in the other set. Figure 5.4(b) states that each entity in set X is related to at most one element in set Y and that each entity in set Y is related to exactly one entity in set X.

The top portion of Figure 5.5 on page 191 asserts that, for any valid index i, there is a one-to-one relationship between the elements of the sets $\tilde{D}_i^{(N)}$ and $\tilde{G}_i^{(N)}$. More precisely, it asserts that each element in the two sets maps to exactly one in the other set. For $i \in \tau(N)$, the dpwcm function maps 4-tuples from a $\tilde{D}_i^{(N)}$ set to generally qualifying 4-compositions in a $\tilde{G}_i^{(N)}$ set (the value of i is the same for both sets). Table 5.3 on page 160 and Figure 5.2 on page 179 have examples of this kind of mapping.

The figure also asserts that, for some set $\{i_1, i_2, \ldots, i_j\}$ of valid indices, the intersection of the various $\tilde{G}_i^{(N)}$ sets is the intersection set

$$\tilde{G}_{i_1}^{(N)} \cap \tilde{G}_{i_2}^{(N)} \cdots \cap \tilde{G}_{i_j}^{(N)}$$

and that there is a one-to-one relationship from an individual $\tilde{G}_i^{(N)}$ to the intersection set. The reason for this is that the intersection set contains only those elements that are in *each* of $\tilde{G}_{i_1}^{(N)}, \tilde{G}_{i_2}^{(N)}, \cdots$, and $\tilde{G}_{i_j}^{(N)}$. Therefore, any element of an individual $\tilde{G}_i^{(N)}$, where $i \in \{i_1, i_2, \ldots, i_j\}$, that is not also an element of every other $\tilde{G}_j^{(N)}$, where $j \in \{i_1, i_2, \ldots, i_j\}$, but $j \neq i$, for the specified index set, does not map to any element of the intersection set. If such an element is represented in all of the other $\tilde{G}_i^{(N)}$, for the specified index set, then it maps to exactly one element in the intersection set. Conversely, any element in the intersection set is always guaranteed to map to exactly one element in each of the individual $\tilde{G}_i^{(N)}$ sets because the elements in the intersection set are those that the individual sets have in common.

The bottom portion of Figure 5.5 on page 191 is related to the concepts that were introduced by Definitions 5.8.4.3 to 5.8.4.7 on pages 165–166, inclusive. Essentially, these definitions are multiple index extensions of these three concepts: a divisor pair, a set of 4-tuple representations of a divisor pair, and a 4-tuple generally qualifying composition of a divisor pair.

The main significance of this bottom portion is that it is not necessary to know the elements of the set

$$\tilde{G}_{i_1}^{(N)} \cap \tilde{G}_{i_2}^{(N)} \cdots \cap \tilde{G}_{i_j}^{(N)}$$

in order to determine its cardinality. The cardinality can be determined by the properties of the associated divisor pairs for a specified index set. All that is necessary is this sequence of steps: calculate the column-wise greatest common divisor pair g of the associated divisor pairs, convert g to its 4-tuple representation, and count its number of elements. This can also be determined analytically by calculating the number of weak 4-compositions for the first component of g and also for its second component. The product of these two numbers is the same as the cardinality of $\tilde{D}_{i_1,i_2,\ldots,i_j}^{(N)}$. Note that Figure 5.5 on the next page also asserts that there is a bijection between any two of the sets

$$\begin{aligned}&\tilde{G}_{i_1}^{(N)} \cap \tilde{G}_{i_2}^{(N)} \cdots \cap \tilde{G}_{i_j}^{(N)},\\ &\tilde{D}_{i_1,i_2,\ldots,i_j}^{(N)}, \text{ and}\\ &\tilde{G}_{i_1,i_2,\ldots,i_j}^{(N)}.\end{aligned}$$

5.8.7 Running Example: Intersection of Three Sets of Generally Qualifying Weak 4-Compositions

In the discussion to follow, assume that $N = 8$. Table 5.2 on page 158 lists the four divisor pairs that are possible for an N with this value. For the convenience of the reader, these pairs are repeated below. A positive integer N is related to one or more divisor pairs by an "integer to divisor pairs" relationship. This relationship is one-to-many from the set of positive integers to the set of divisor pairs. The set $T^{(N)}$ contains exactly the divisor pairs for N and the sets $D^{(N)}$ and $\tilde{D}^{(N)}$ contain the corresponding 4-tuple representations for 4-compositions and weak 4-compositions, respectively. The sets $D_i^{(N)}$ and $\tilde{D}_i^{(N)}$, respectively, are derived from the sets $D^{(N)}$ and $\tilde{D}^{(N)}$.

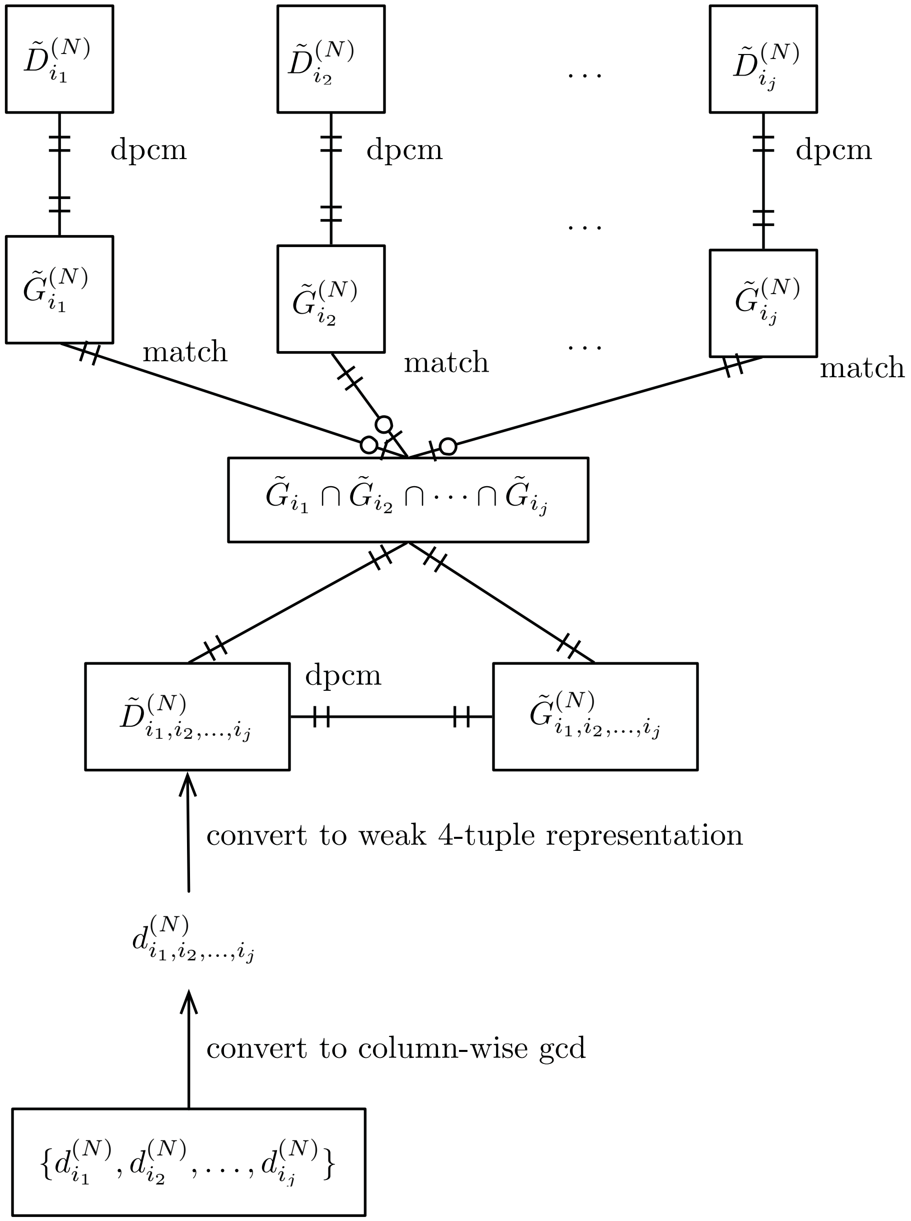

$\tilde{D}_{i_1}^{(N)}$
$\tilde{D}_{i_2}^{(N)}$
$\cdots$
$\tilde{D}_{i_j}^{(N)}$
dpcm
dpcm
dpcm
$\tilde{G}_{i_1}^{(N)}$
$\tilde{G}_{i_2}^{(N)}$
$\cdots$
$\tilde{G}_{i_j}^{(N)}$
match
match
$\cdots$
match
$\tilde{G}_{i_1} \cap \tilde{G}_{i_2} \cap \cdots \cap \tilde{G}_{i_j}$
dpcm
$\tilde{D}_{i_1,i_2,\ldots,i_j}^{(N)}$
$\tilde{G}_{i_1,i_2,\ldots,i_j}^{(N)}$
convert to weak 4-tuple representation
$d_{i_1,i_2,\ldots,i_j}^{(N)}$
convert to column-wise gcd
$\{d_{i_1}^{(N)}, d_{i_2}^{(N)}, \ldots, d_{i_j}^{(N)}\}$

Figure 5.5: ER Diagram of the Main Relationships.

For example, the positive integer 8 is related to four divisor pairs (i.e., (1, 8), (2, 4), (4, 2), (8, 1)). These are precisely the divisor pairs that are enumerated in Table 5.2 on page 158. The indices of these divisor pairs start at 1 and end at 4 as the table cells in the bottom row are visited in a left to right manner. By this information, note that index i has the value 1 for the $(1, 8)$ pair and that it has has the value 4 for the $(8, 1)$ pair.

The $\tilde{D}_i^{(N)}$ set contains the mapped 4-tuples for divisor pair i. One example that corresponds to a feasible mapping is this one: $d_2^{(8)} = (2, 4)$ and

$$\begin{aligned}\tilde{D}_2^{(8)} =\{ & \\ & (0,2,0,4),(0,2,1,3),(0,2,2,2),(0,2,3,1),(0,2,4,0), \\ & (1,1,0,4),(1,1,1,3),(1,1,2,2),(1,1,3,1),(1,1,4,0), \\ & (2,0,0,4),(2,0,1,3),(2,0,2,2),(2,0,3,1),(2,0,4,0) \\ \}. &\end{aligned}$$

The Venn diagram in Figure 5.3 on page 186 depicts the intersection of three sets of generally qualifying weak 4-compositions. The index set for these three sets is $\{1, 2, 3\}$. Therefore, the sets of generally qualifying weak 4-compositions are $\tilde{G}_1^{(8)}$, $\tilde{G}_2^{(8)}$, and $\tilde{G}_3^{(8)}$. These sets correspond to those for divisor pairs $(1, 8), (2, 4)$, and $(4, 2)$, respectively.

The divisor pairs for the indices have been rewritten in terms of the common gcd for each component of the pairs. The common gcd for the first component of each pair is

$$\gcd(d_1^{(8)}[1], d_2^{(8)}[1], d_3^{(8)}[1]) = \gcd(1, 2, 4) = 1$$

and is

$$\gcd(d_1^{(8)}[2], d_2^{(8)}[2], d_3^{(8)}[2]) = \gcd(8, 4, 2) = 2$$

for the second component of each pair. This is evidenced in the multiplicand of each of the rewritten divisor pairs below.

$$((d_1^{(8)}[1]/1)1, (d_1^{(8)}[2]/2)) = (1 \cdot 1, 4 \cdot 2)$$
$$((d_2^{(8)}[1]/1)1, (d_1^{(8)}[2]/2)) = (2 \cdot 1, 2 \cdot 2)$$
$$((d_3^{(8)}[1]/1)1, (d_1^{(8)}[2]/2)) = (4 \cdot 1, 1 \cdot 2)$$

The corresponding multipliers are used to construct the three mapping functions below. Notice that the multiplier for the first component of a divisor pair is also the multiplier for the first two variables in its corresponding mapping function and the multiplier for the second component of a divisor pair is the multiplier for the last two variables in its corresponding mapping function.

$$f_1(w, x, y, z) = (1 \cdot w, 1 \cdot x, 4 \cdot y, 4 \cdot z) = (w, x, 4y, 4z)$$
$$f_2(w, x, y, z) = (2 \cdot w, 2 \cdot x, 2 \cdot y, 2 \cdot z) = (2w, 2x, 2y, 2z)$$
$$f_3(w, x, y, z) = (4 \cdot w, 4 \cdot x, 1 \cdot y, 1 \cdot z) = (4w, 4x, y, z)$$

Collectively, the information from these mapping functions indicate that the generally qualifying weak 4-compositions that are in the intersection of sets $\tilde{G}_1^{(8)}$, $\tilde{G}_2^{(8)}$, and $\tilde{G}_3^{(8)}$ must meet all of these conditions: the value of each of the four components of the weak compositions must be evenly divisible by 4 because the least common multiple of the values 1, 2, and 4 is 4. The only weak compositions in the Venn diagram of Figure 5.3 on page 186 that meet this condition are the ones that are in the intersection of the three sets.

Table 5.6: The Divisor Pairs for $N = 8$ and Their Associated Sets.

$\tilde{D}^{(8)}_{1,2,3}$	$L^{(8)}_1$	$L^{(8)}_2$	$L^{(8)}_3$	$\tilde{I}'$	$M^{(8)}_1, M^{(8)}_2, M^{(8)}_3$
$(0, 1, 0, 2)$	$(0, 1, 0, 8)$	$(0, 2, 0, 4)$	$(0, 4, 0, 2)$	$(0, 2, 0, 0)$	$(0, 8, 0, 0)$
$(0, 1, 1, 1)$	$(0, 1, 4, 4)$	$(0, 2, 2, 2)$	$(0, 4, 1, 1)$	$(0, 1, 0, 1)$	$(0, 4, 0, 4)$
$(0, 1, 2, 0)$	$(0, 1, 8, 0)$	$(0, 2, 4, 0)$	$(0, 4, 2, 0)$	$(0, 0, 0, 2)$	$(0, 0, 0, 8)$
$(1, 0, 0, 2)$	$(1, 0, 0, 8)$	$(2, 0, 0, 4)$	$(4, 0, 0, 2)$	$(0, 0, 2, 0)$	$(0, 0, 8, 0)$
$(1, 0, 1, 1)$	$(1, 0, 4, 4)$	$(2, 0, 2, 2)$	$(4, 0, 1, 1)$	$(1, 0, 1, 0)$	$(4, 0, 4, 0)$
$(1, 0, 2, 0)$	$(1, 0, 8, 0)$	$(2, 0, 4, 0)$	$(4, 0, 2, 0)$	$(2, 0, 0, 0)$	$(8, 0, 0, 0)$

5.9 Calculating $\mathcal{Q}'_{\text{CLM}}$ for a Document Collection of Size N

The proof of Lemma 5.8.5 provides a closed form expression to calculate the number of generally qualifying weak 4-compositions for j divisor pairs. This expression is used below in the proof of Lemma 5.9.1. The proof of Lemma 5.9.2 on page 196 uses the results of Lemma 5.9.1 to provide an equation that calculates the total number of generally qualifying 4-compositions for a document collection of size N. Figure 5.6 on the next page depicts the situation that is discussed in this section.

Lemma 5.9.1. *Suppose $\tilde{G}^{(N)}_1$, $\tilde{G}^{(N)}_2$, $\ldots$, $\tilde{G}^{(N)}_m$, where $m = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of m (i.e., $1 \leq i_1 < i_2 < \cdots < i_j \leq m$), are the sets of generally qualifying weak 4-compositions for a document collection of size N. Then*

$$\left| \bigcup_{1 \leq i \leq m} \tilde{G}^{(N)}_i \right| = \sum_{j=1}^{m} (-1)^{j-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq m} |\tilde{G}^{(N)}_{i_1} \cap \tilde{G}^{(N)}_{i_2} \cdots \cap \tilde{G}^{(N)}_{i_j}| \tag{5.9.1}$$

$$= \sum_{j=1}^{m} (-1)^{j-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq m} (d^{(N)}_{i_1, i_2, \ldots, i_j}[1] + 1)(d^{(N)}_{i_1, i_2, \ldots, i_j}[2] + 1). \tag{5.9.2}$$

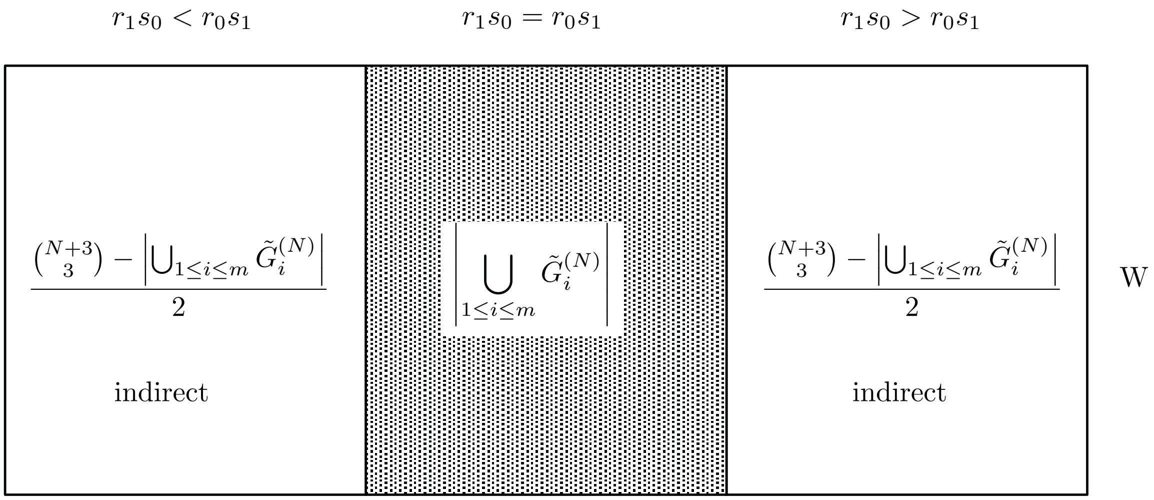

$r_1 s_0 < r_0 s_1$
$r_1 s_0 = r_0 s_1$
$r_1 s_0 > r_0 s_1$
$\frac{\binom{N+3}{3} - \left|\bigcup_{1 \leq i \leq m} \tilde{G}_i^{(N)}\right|}{2}$
indirect
$\left|\bigcup_{1 \leq i \leq m} \tilde{G}_i^{(N)}\right|$
$\frac{\binom{N+3}{3} - \left|\bigcup_{1 \leq i \leq m} \tilde{G}_i^{(N)}\right|}{2}$
indirect
W

Figure 5.6: This figure corresponds to the discussion in Section 5.9. It is assumed that the document collection is non-empty (i.e., $N \geq 1$). The number of weak 4-compositions that satisfy the restriction $r_1 s_0 > r_0 s_1$ can be determined if there is a method to calculate the number of weak 4-compositions that satisfy $r_1 s_0 = r_0 s_1$. The former value is calculated by subtracting the latter one from the value for the cardinality of W (the number of weak 4-compositions for N) and then dividing the result by 2. The gray area indicates that the value for $r_1 s_0 = r_0 s_1$ is directly calculated whereas the white areas indicate that the value for $r_1 s_0 > r_0 s1$ and $r_1 s_0 < r_0 s_1$ are indirectly calculated. The symbol W represents the set of weak 4-compositions for N.

Proof. Lemma 5.8.3 on page 172 and Lemma 5.8.5 on page 181 enable the rewriting of Equation 5.9.1 on page 194 as Equation 5.9.2 on page 194. □

Lemma 5.9.2.

$$\textit{The contribution is} \begin{cases} \dfrac{\tilde{C}_4(N) - \left|\bigcup_{1\le i\le m} \tilde{G}_i^{(N)}\right|}{2}, & \textit{if } N \geq 1; \\ 0, & \textit{otherwise;} \end{cases}$$

when the condition $p' > t'$ is true.

Proof. The expression

$$\bigcup_{1\le i\le m} \tilde{G}_i^{(N)}$$

calculates the number of weak 4-compositions of $N \geq 0$ where $r_1 s_0 = r_0 s_1$. By Lemma 5.2.1 on page 146, the number of weak 4-compositions of N that satisfy $r_1 s_0 > r_0 s_1$ is the same as the number of weak 4-compositions of N that satisfy $r_1 s_0 < r_0 s_1$. Therefore, the number of weak 4-compositions of N that satisfy $p' > t'$ is

$$\frac{\tilde{C}_4(N) - \left|\bigcup_{1\le i\le m} \tilde{G}_i^{(N)}\right|}{2}.$$

By Equation 2.2.2 on page 26, the cardinality of the sample space of weak 4-compositions for N is

$$\binom{N+3}{3}.$$

□

After dividing the former expression by the latter expression, we obtain

$$\mathcal{Q}'_{\text{CLM}} = \Pr(p' > t') = \begin{cases} \frac{\tilde{C}_4(N) - \left|\bigcup_{1\le i\le m} \tilde{G}_i^{(N)}\right|}{2\binom{N+3}{3}}, & \text{if } N \geq 1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.9.3}$$

5.10 A Refinement of the Calculations for $\mathcal{Q}'_{\text{CLM}}$

An alternate way to derive a POIE-based equation for $\mathcal{Q}'_{\text{CLM}}$ is to make use of several of the closed form expressions that were developed, and verified, in Chapter 4. These expressions count the number of weak 4-compositions that satisfy the relation $p' > t'$ for an N document collection in all situations, except where the conditions $p \in (0,1)$ and $q \in (0,1)$ are both true. These expressions correspond to the situation where the value of at least one component of every weak 4-tuple is 0. Figure 5.7 depicts the situation that is discussed in this section.

	$r_1s_0 < r_0s_1$	$r_1s_0 = r_0s_1$	$r_1s_0 > r_0s_1$	
Quadrant I	0	$N+1$	0	W_1
Quadrant II	0	0	0	W_2
Quadrant III	0	$N+1$	0	W_3
Quadrant IV	$2\binom{N-1}{2} + N - 1$	$\binom{N+3}{3} - \binom{N-1}{3} - 4\left(N + \binom{N-1}{2}\right)$	$2\binom{N-1}{2} + N - 1$	$W_4 \backslash C_4$
	$\dfrac{\binom{N-1}{3} - \left\lvert\bigcup_{1\le i\le m} G_i^{(N)}\right\rvert}{2}$ indirect	$\left\lvert\bigcup\limits_{1\le i\le m} G_i^{(N)}\right\rvert$	$\dfrac{\binom{N-1}{3} - \left\lvert\bigcup_{1\le i\le m} G_i^{(N)}\right\rvert}{2}$ indirect	C_4

Figure 5.7: This figure corresponds to the discussion in Section 5.10. It is assumed that the document collection is non-empty (i.e., $N \geq 1$). The cells that do not have a gray background, nor are labeled indirect, contain values that were determined by the use of the equations from Table 4.11 on page 140 for determining the number of weak 4-compositions in Quadrants I, II, and III, plus the equation for the number of weak 4-compositions in Quadrant IV that are not also strong compositions. The gray area represents the value that needs to be calculated so that the number of 4-compositions in Quadrant IV that satisfy the restriction $r_1s_0 > r_0s_1$ can be indirectly calculated. The Ws in this figure represent weak 4-compositions and the Cs represent strong 4-compositions. More specifically, W_1, W_2, W_3, and W_4, respectively, represent the number of weak 4-compositions for Quadrants I, II, III, and IV. The symbol C_4 represents the set of strong compositions for Quadrant IV. The expression $W_4 \backslash C_4$ represents the set of weak 4-compositions in Quadrant IV that are not simultaneously strong compositions.

We proceed in two stages. The first stage develops the total count for each of the four quadrants, except for those weak 4-compositions that satisfy both $p \in (0, 1)$ and $q \in (0, 1)$. The second stage develops the count just for the part of Quadrant IV that was not covered by the expressions that were developed in the previous chapter. These weak 4-compositions in the second stage correspond to those that satisfy both $p \in (0, 1)$ and $q \in (0, 1)$. These weak 4-compositions are also 4-compositions because the value of each of their four components is a positive integer.

Part of the discussion in Chapter 4 indicated that we could separate the problem of determining the count contributions into four subproblems. There is a one-to-one correspondence between the set of subproblems and the set of quadrants. Each subproblem is concerned with finding the count contribution for the quadrant that it maps to. Once we find this count for each of the quadrants, we total the counts. The result is the count contribution for the original problem.

In order to determine the count contributions for this (i.e., CLM) ranking method, we start by first developing the expressions that count the number of qualifying weak compositions for Quadrants I, II, and III. After, that we do the same for all of the categories of Quadrant IV, except for the category where both $p' \in (0, 1)$ and $q' \in (0, 1)$ hold. Lastly, we develop the count contribution expressions for this remaining part of Quadrant IV.

5.10.1 The Number of Qualifying Weak Compositions for Quadrants I, II, and III

How do we determine the contribution count (i.e., the number of qualifying weak 4-compositions), when $p' > t'$ is true, and the document collection is non-empty, for these three quadrants? The results of the analyses from Chapter 4 provide the answer. From the information in Table 4.11 on page 140, the counts for the first three quadrants are 0, 0, and 0, respectively, for a combined count of 0.

From an information retrieval perspective, with query q, and a document collection of size N, the weak 4-compositions that comprise Quadrant I correspond to the situation where every document in the collection is relevant and the collection has at least one document (i.e., $s_0 + s_1 = 0$ and $r_0 + r_1 > 0$). The weak 4-compositions for Quadrant II correspond to an empty collection (i.e., $s_0 + s_1 = 0$ and $r_0 + r_1 = 0$). And, the weak 4-compositions for Quadrant III correspond to the situation where every document in the collection is non-relevant and the collection has at least one document (i.e., $s_0 + s_1 > 0$ and $r_0 + r_1 = 0$).

Lemma 5.10.1. *The total contribution count is 0 when $p' > t'$ holds.*

Proof. The total contribution is the sum of the values in column 5 of lines 1–3, inclusive, in Table 4.11 on page 140. It indicates that the count contributions for each of Quadrants I, II, and III is 0 when $p' > t'$ is true. Their collective total is 0. □

5.10.2 The Number of Qualifying Weak Compositions for Quadrant IV (each weak 4-composition in this quadrant represents a document collection that has positive numbers of relevant and non-relevant documents) When At Least One of the Parameters r_1, r_0, s_1, and s_0 Has a Value of Zero

From an information retrieval perspective, with query q, and a document collection of size N, the weak 4-compositions that comprise Quadrant IV correspond to the situation where both the number of relevant and the number of non-relevant documents are positive (i.e., $s_0 + s_1 > 0$ and $r_0 + r_1 > 0$). In this particular section, the total contribution count is for all situations in Quadrant IV, except for those situations where each of the four parameters in a weak 4-composition (r_1, s_0, r_0, s_1) has a positive (i.e., greater than zero) value. The counts for these latter situations are addressed in Section 5.10.3.

The count for this section can also be determined solely from the information in Table 4.11 on page 140. The following lemma addresses the value for this count.

Lemma 5.10.2.

$$\textit{The count contribution is } \begin{cases} 2\binom{N-1}{2} + N - 1, & \textit{if } N \geq 1; \\ 0, & \textit{otherwise;} \end{cases}$$

when $p' > t'$ holds.

Proof. If $p = 0$, then $r_1 = 0$ is true; if $p = 1$, then $r_0 = 0$ is true; if $q = 0$, then $s_1 = 0$ is true; and if $q = 1$, then $s_0 = 0$ is true because, from the discussions in Chapter 4,

$$\begin{aligned} p = 0 &\Longrightarrow \frac{r_1}{r_1 + r_0} = 0 \Longrightarrow r_1 = 0, \\ p = 1 &\Longrightarrow \frac{r_1}{r_1 + r_0} = 1 \Longrightarrow r_0 = 0, \\ q = 0 &\Longrightarrow \frac{s_1}{s_1 + s_0} = 0 \Longrightarrow s_1 = 0, \text{ and} \\ q = 1 &\Longrightarrow \frac{s_1}{s_1 + s_0} = 1 \Longrightarrow s_0 = 0. \end{aligned}$$

From the above implications, and the information in Table 4.11 on page 140, we can see that at least one of the values for r_1, r_0, s_1, and s_0 is 0 for eight of the nine mutually exclusive joint conditions for Quadrant IV that are listed in column 5 of this table. An inspection of this table reveals that all of the supplemental conditions in Table 4.11 on page 140 for lines 4-7, inclusive, and lines 9-12, inclusive, have at least one conjunct where either $p = 0$, $p = 1$, $q = 0$, or $q = 1$ is true. These conditions cover all of the Quadrant IV conditions, except for the one where $p \in (0, 1)$ and $q \in (0, 1)$ are both true.

The sum that corresponds to the 8 conditions is

$$N-1+2\binom{N-1}{2},$$

which is simply the aggregate of the quantities that appear in column 5 of Table 4.11 on page 140 for lines 4–7, inclusive, and lines 9–12, inclusive. The reasoning behind its derivation follows this sentence. From Table 4.11 on page 140, we see that when $N \geq 2$ holds, the partial sum of the contributions is

$$0+0+(N-1)+0=N-1.$$

Additionally, when $N \geq 3$ also holds, we must add

$$0+\binom{N-1}{2}+0+\binom{N-1}{2}=2\binom{N-1}{2}$$

to that value because $N \geq 3$ implies $N \geq 2$. The resultant sum is

$$N-1+2\binom{N-1}{2},$$

and its value is valid even when $N = 1$ or $N = 2$ because the expression

$$\binom{N-1}{2}$$

vanishes (i.e., has the value 0) when $N \in \mathbb{Z}^+$ and $1 \leq N \leq 2$ is true. □

5.10.3 The Number of Qualifying Weak Compositions for Quadrant IV (each weak 4-composition in this quadrant represents a document collection that has positive numbers of relevant and non-relevant documents) When Each of the Parameters r_1, r_0, s_1, and s_0 Has a Positive Value

This section is concerned with determining the count contribution for Quadrant IV when the conditions $p \in (0, 1)$, $q \in (0, 1)$, and $p' > t'$ all hold. When the first and second conditions hold, the values of r_1, r_0, s_1, and s_0 are all positive. This becomes important in the discussion below and in those discussions that appear in later chapters.

From an information retrieval perspective, with query q, and a document collection of size N, the weak 4-compositions that comprise the part of Quadrant IV that is the focus of this section correspond to the situation where both the number of relevant and the number of non-relevant documents are positive (i.e., $s_0 + s_1 > 0$ and $r_0 + r_1 > 0$) and both $p \in (0, 1)$ and $q \in (0, 1)$ are true. The counts for this situation cannot be determined from the information in Table 4.11 on page 140 because the mathematics and arguments needed to determine these counts are considerably more involved than any of the mathematics and arguments that appeared in Chapter 4. The derivation of formulas and techniques that help in determining these counts are the subject of much of the remainder of this chapter.

Two lemmas appear below. The first of them is associated with the situation where the values of p and q are strictly between 0 and 1. The first lemma (i.e., Lemma 5.10.3 on the following page) proves that the number of documents that is associated with each of the four parts of the corresponding 4-compositions must be a positive number. The second lemma (i.e., Lemma 5.10.4 on the next page) proves that when the values of p and q are strictly between 0 and 1, then $p = p'$ is true and $q = q'$ is true. The results from these lemmas are used in several places in this dissertation.

Lemma 5.10.3. *Suppose $p \in (0,1)$ and $q \in (0,1)$ are true. Then r_1, r_0, s_1, and s_0 must all be positive values.*

Proof. From Figure 4.1 on page 115, it is evident that both $r_1 + r_0 > 0$ and $s_1 + s_0 > 0$ must hold for any outcome that is a member of Quadrant IV. The expression

$$p = \frac{r_1}{r_1 + r_0} \in (0,1)$$

implies that the conditions $r_1 > 0$ and $r_0 > 0$ hold because the value of r_1 must be positive in order for p to be positive, but that the value of r_0 must be positive, also, so that the value of p cannot equal or exceed the value 1.

The argument for q is similar to the one above for p. The expression

$$q = \frac{s_1}{s_1 + s_0} \in (0,1)$$

implies that the conditions $s_1 > 0$ and $s_0 > 0$ hold because the value of s_1 must be positive in order for q to be positive, but that the value of s_0 must be positive, also, so that the value of q is always less than the value 1. □

Lemma 5.10.4. *Suppose $p \in (0,1)$ and $q \in (0,1)$ are true. Then $p = p'$ and $q = q'$ are also true.*

Proof. This trivially follows from the definitions of p' and q' on page 120 in Section 4.3. The definition of p' states that $p' = p$ when $0 < p < 1$ and the analogous definition for q' states that $q' = q$ when $0 < q < 1$. It is well-known that, for a real value x, such as those represented by p and q, the expressions $0 < x < 1$ and $x \in (0,1)$ are equivalent. □

By Lemma 5.10.3, because each of r_1, r_0, s_1, and s_0 is positive, the weak 4-compositions in this section are also 4-compositions. Therefore, in the remainder of this section, we use 4-compositions, the more specific term. The use of this term is not possible either for Quadrants I, II, and III, or for all of the other 8 conditions for p and q that are listed in Table 4.11 on page 140 for this quadrant, because at least one of the parameters r_1, r_0, s_1, and s_0 in all of those situations is guaranteed to have a value of 0. Note that, if any component of a 4-tuple, that consists of all natural numbers, is 0, then this 4-tuple cannot possibly be a strong 4-composition; it can only be a weak 4-composition.

The Four Cases For This Part of Quadrant IV

The calculations for this part of Quadrant IV can be broken down into several cases: the four component values are identical; only two distinct values occur among the 4 component values; only three distinct values occur among the 4 component values; and, lastly, all the component values are unique. The sections below derive expressions for the contribution that each of these cases make to the overall total.

Unlike the solutions to Quadrants I, II, III, and all categories of Quadrant IV, except for the one where $p \in (0, 1)$ and $q \in (0, 1)$, the solutions to this category cannot be expressed as a closed formula. An algorithm is developed for each one. The algorithms, in both cases, rely on integer factorization properties of N (Rosen, 2005) and the Principle of Inclusion-Exclusion (Stanley, 1997; Rosen et al., 2000; Charalambides, 2002; Bóna, 2006; Lovász, 2007; Bóna, 2007).

Lemma 5.10.5. *Suppose $r = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of r (i.e., $1 \leq i_1 < i_2 < \cdots < i_j \leq r$). Then the cardinality of the set $G_{i_1}^{(N)} \cap G_{i_2}^{(N)} \cap \cdots \cap G_{i_j}^{(N)}$ is $(c_1 - 1)(c_2 - 1)$ where $c_1 = d_{i_1,i_2,\ldots,i_j}^{(N)}[1]$ and $c_2 = d_{i_1,i_2,\ldots,i_j}^{(N)}[2]$.*

Proof. The proof for this lemma is very similar to that of Lemma 5.8.3 on page 172. The essential difference is that this lemma is concerned with strong compositions rather than weak compositions. Therefore, the expression that calculates the cardinality is based on the strong 2-compositions of $c_1 = d_{i_1,i_2,\ldots,i_j}^{(N)}[1]$ and $c_2 = d_{i_1,i_2,\ldots,i_j}^{(N)}[2]$, rather than with the weak 2-compositions, as was the situation with Lemma 5.8.3 on page 172. By Equation 2.2.1 on page 26, the number of strong 2-compositions for c_1 is

$$C_2(c_1) = \binom{c_1 - 1}{2 - 1} = \binom{c_1 - 1}{1} = c_1 - 1$$

and the number of strong 2-compositions for c_2 is

$$C_2(c_2) = \binom{c_2 - 1}{2 - 1} = \binom{c_2 - 1}{1} = c_2 - 1.$$

Hence, the cardinality of the set $G_{i_1}^{(N)} \cap G_{i_2}^{(N)} \cap \cdots \cap G_{i_j}^{(N)}$ is $(c_1 - 1)(c_2 - 1)$. □

Lemma 5.10.6. *Suppose $G_1^{(N)}$, $G_2^{(N)}$, $\ldots$, $G_m^{(N)}$, where $m = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of m (i.e., $1 \leq i_1 < i_2 < \cdots < i_j \leq m$), are the sets of qualifying 4-compositions with mutually distinct components. Then*

$$\left| \bigcup_{1 \leq i \leq m} G_i^{(N)} \right| = \sum_{j=1}^{m} (-1)^{j-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq m} |G_{i_1}^{(N)} \cap G_{i_2}^{(N)} \cdots \cap G_{i_j}^{(N)}| \tag{5.10.1}$$

$$= \sum_{j=1}^{m} (-1)^{j-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq m} (d_{i_1, i_2, \ldots, i_j}^{(N)}[1] - 1)(d_{i_1, i_2, \ldots, i_j}^{(N)}[2] - 1). \tag{5.10.2}$$

Proof. Lemma 5.10.5 on the previous page enables the rewriting of Equation 5.10.1 as Equation 5.10.2. □

After putting all of this together, we obtain

$$\mathcal{Q}'_{\text{CLM}} = \Pr(p' > t') = \begin{cases} \dfrac{2\binom{N-1}{2} + N - 1 + \frac{C_4(N) - \left| \bigcup_{1 \leq i \leq m} G_i^{(N)} \right|}{2}}{\binom{N+3}{3}}, & \text{if } N \geq 1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.10.3}$$

5.11 A Further Refinement of the Calculations for $\mathcal{Q}'_{\mathrm{CLM}}$

An alternate way to derive a POIE equation for the number of generally qualifying 4-compositions is to break the task of determining this number into mutually exclusive parts and, later, combine the results from these parts. There are two major benefits to this: the primary one is that it provides additional validation of the proof for Lemma 5.10.6 on the preceding page; the secondary benefit it that it provides some distributional information about the number of qualifying 4-compositions that are associated with each part. This additional information provides more insight about the compositions. Figure 5.8 on the next page depicts the situation that is discussed in this section.

The four scenarios that are discussed in subsections 5.11.1, 5.11.2, 5.11.3, and 5.11.4 are based on how many unique values there are among those assigned to variables r_1, s_0, r_0, and s_1. These scenarios correspond to exactly 1 distinct value, exactly 2 distinct values, exactly 3 distinct values, and exactly 4 distinct values, respectively.

Lemma 5.11.1. *Suppose $G_1^{(N)}$, $G_2^{(N)}$, $\ldots$, $G_m^{(N)}$, where $m = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of m (i.e., $1 \le i_1 < i_2 < \cdots < i_j \le m$), are the sets of qualifying 4-compositions. Then the number of generally qualifying 4-compositions for these j indices is*

$$\begin{aligned}
|G_{i_1}^{(N)} \cap G_{i_2}^{(N)} \cdots \cap G_{i_j}^{(N)}| &= \left|G_{i_1,i_2,\ldots,i_j}^{(N)}\right| \\
&= \left|G_{1;i_1,i_2,\ldots,i_j}^{(N)}\right| + \left|G_{2;i_1,i_2,\ldots,i_j}^{(N)}\right| + \\
&\quad \left|G_{3;i_1,i_2,\ldots,i_j}^{(N)}\right| + \left|G_{4;i_1,i_2,\ldots,i_j}^{(N)}\right| \\
&= (d_{i_1,i_2,\ldots,i_j}^{(N)}[1] - 1)(d_{i_1,i_2,\ldots,i_j}^{(N)}[2] - 1),
\end{aligned}$$

where $\left|G_{1;i_1,i_2,\ldots,i_j}^{(N)}\right|$, $\left|G_{2;i_1,i_2,\ldots,i_j}^{(N)}\right|$, $\left|G_{3;i_1,i_2,\ldots,i_j}^{(N)}\right|$, and $\left|G_{4;i_1,i_2,\ldots,i_j}^{(N)}\right|$ represent the one, two,

	$r_1s_0 < r_0s_1$	$r_1s_0 = r_0s_1$	$r_1s_0 > r_0s_1$	
Quadrant I	0	$N+1$	0	W_1
Quadrant II	0	0	0	W_2
Quadrant III	0	$N+1$	0	W_3
Quadrant IV	$2\binom{N-1}{2}+N-1$	$\binom{N+3}{3}-\binom{N-1}{3}-4\left(N+\binom{N-1}{2}\right)$	$2\binom{N-1}{2}+N-1$	$W_4\backslash C_4$
	$\frac{\binom{N-1}{3}-\left\lvert\bigcup_{1\leq i\leq m} G_i^{(N)}\right\rvert}{2}$ indirect	$\left\lvert\bigcup\limits_{1\leq i\leq m} G_i^{(N)}\right\rvert$	$\frac{\binom{N-1}{3}-\left\lvert\bigcup_{1\leq i\leq m} G_i^{(N)}\right\rvert}{2}$ indirect	C_4

Figure 5.8: This figure corresponds to the discussion in Section 5.11. The cells that do not have a gray background, nor are labeled indirect, contain values that were determined by the use of the equations from Table 4.11 on page 140 for determining the number of weak 4-compositions in Quadrants I, II, and III, plus the equation for the number of weak 4-compositions in Quadrant IV that are not also strong compositions. The gray area represents the values that need to be calculated so that the number of 4-compositions in Quadrant IV that satisfy the restriction $r_1s_0 > r_0s_1$ can be indirectly calculated. The essential difference between the situation that is being depicted with this figure and that of Figure 5.7 on page 197 is that the gray region for $r_1s_0 = r_0s_1$ is divided into four non-overlapping parts. The count contribution for each part is determined, then added to form a total that is then used to indirectly calculate the number of 4-compositions in Quadrant IV that satisfy the restriction $r_1s_0 > r_0s_1$. The Ws in this figure represent weak 4-compositions and the Cs represent strong 4-compositions. More specifically, W_1, W_2, W_3, and W_4, respectively, represent the number of weak 4-compositions for Quadrants I, II, III, and IV. The symbol C_4 represents the set of strong compositions for Quadrant IV. The expression $W_4\backslash C_4$ represents the set of weak 4-compositions in Quadrant IV that are not simultaneously strong compositions.

three, and four distinct values scenarios, respectively.

Proof. The proof is divided into four parts, one for the number of distinct values in each of the four scenarios. There is a lemma and associated proof for each of these scenarios. These lemmas (i.e., Lemma 5.11.1 on page 211, Lemma 5.11.2 on page 213, Lemma 5.11.3 on page 218, Lemma 5.11.4 on page 222) and their proofs follow this one and are in the next subsections. The proof for this lemma consists of summing the counting expressions that are associated with these 4 lemmas and showing that their total value is equal to

$$(d^{(N)}_{i_1,i_2,\ldots,i_j}[1]-1)(d^{(N)}_{i_1,i_2,\ldots,i_j}[2]-1),$$

which is identical to the value that is associated with the $|G^{(N)}_{i_1} \cap G^{(N)}_{i_2} \cdots \cap G^{(N)}_{i_j}|$ expression in Lemma 5.10.5 on page 204.

From subsections 5.11.1, 5.11.2, 5.11.3, and 5.11.4, we obtain these equations:

$$\begin{aligned}
|G^{(N)}_{1;i_1,i_2,\ldots,i_j}| &= [d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \text{ is even}] \times [d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \text{ is even}],\\
|G^{(N)}_{2;i_1,i_2,\ldots,i_j}| &= 2 \times (x\,[d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \text{ is even}] + y\,[d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \text{ is even}]),\\
|G^{(N)}_{3;i_1,i_2,\ldots,i_j}| &= 4 \times \lfloor (\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2 \rfloor, \text{ and}\\
|G^{(N)}_{4;i_1,i_2,\ldots,i_j}| &= |G'^{(N)}_{4;i_1,i_2,\ldots,i_j}| - 4 \times \lfloor (\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2 \rfloor,
\end{aligned}$$

where

$$\begin{aligned}
|G'^{(N)}_{4;i_1,i_2,\ldots,i_j}| = (&d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - 1 - [d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \text{ is even}])\times\\
&(d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - 1 - [d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \text{ is even}]),\\
x &= \lfloor (d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - 1)/2 \rfloor, \text{ and}\\
y &= \lfloor (d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - 1)/2 \rfloor.
\end{aligned}$$

In order to simplify the notation that is used in the remainder of this proof, let

$$\begin{aligned}
A &= [d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \text{ is even}],\\
B &= [d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \text{ is even}],\\
C &= \lfloor (\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2 \rfloor,\\
d_1 &= d^{(N)}_{i_1,i_2,\ldots,i_j}[1], \text{and}\\
d_2 &= d^{(N)}_{i_1,i_2,\ldots,i_j}[2].
\end{aligned}$$

Then the equations for the 4 lemmas can be rewritten as

$$\begin{aligned}
|G^{(N)}_{1;i_1,i_2,\ldots,i_j}| &= AB,\\
|G^{(N)}_{2;i_1,i_2,\ldots,i_j}| &= 2(xB + yA),\\
|G^{(N)}_{3;i_1,i_2,\ldots,i_j}| &= 4C,\\
|G^{(N)}_{4;i_1,i_2,\ldots,i_j}| &= (d_1 - 1 - A)(d_2 - 1 - B) - 4C.
\end{aligned}$$

The sum of the values that are associated with these equations is expressed by the equation

$$\begin{aligned}
\left|G^{(N)}_{i_1,i_2,\ldots,i_j}\right| &= \left|G^{(N)}_{1;i_1,i_2,\ldots,i_j}\right| + \left|G^{(N)}_{2;i_1,i_2,\ldots,i_j}\right| +\\
&\qquad \left|G^{(N)}_{3;i_1,i_2,\ldots,i_j}\right| + \left|G^{(N)}_{4;i_1,i_2,\ldots,i_j}\right|\\
&= (AB) + (2(xB + yA)) + (4C) + ((d_1 - 1 - A)(d_2 - 1 - B) - 4C)\\
&= AB + 2(xB + yA) + (d_1 - 1 - A)(d_2 - 1 - B)\\
&= AB + 2(xB + yA) + (d_1 - 1)(d_2 - 1) - (d_1 - 1)B - (d_2 - 1)A \ + AB\\
&= (d_1 - 1)(d_2 - 1) + 2(AB + xB + yA) - (d_1 - 1)B - (d_2 - 1)A
\end{aligned}$$

$$= (d_1 - 1)(d_2 - 1) + S.$$

Note that $S = 2(AB+xB+yA)-(d_1-1)B-(d_2-1)A$ represents the part of the equation that is *sensitive* to whether the values of d_1 and d_2 are even or odd. In determining the value of S, there are a total of 4 possibilities that must be considered because the value of d_1 can be even or odd, independent of whether the value of d_2 is even or odd. These possibilities are enumerated in Table 5.7.

The information in the table illustrates that the value of S is 0 for each of the four possibilities. Hence,

$$\begin{aligned}(d_1 - 1)(d_2 - 1) + S &= (d_1 - 1)(d_2 - 1) + 0 \\ &= (d_1 - 1)(d_2 - 1) \\ &= |G_{i_1}^{(N)} \cap G_{i_2}^{(N)} \cdots \cap G_{i_j}^{(N)}|.\end{aligned}$$

From this result, we can conclude that the values that are computed by Lemma 5.10.6 on page 205 and this lemma (i.e., Lemma 5.11.1 on page 206) both compute the same value for the expression $|G_{i_1}^{(N)} \cap G_{i_2}^{(N)} \cdots \cap G_{i_j}^{(N)}|$.

Table 5.7: The Four Possibilities for the Evaluation of S.

d_1	d_2	A	B	x	y	$2(AB + xB + yA)$	$(d_1 - 1)B$	$(d_2 - 1)A$	S
even	even	1	1	$\frac{d_1-2}{2}$	$\frac{d_2-2}{2}$	$d_1 + d_2 - 2$	$d_1 - 1$	$d_2 - 1$	0
even	odd	1	0	$\frac{d_1-2}{2}$	$\frac{d_2-1}{2}$	$d_2 - 1$	0	$d_2 - 1$	0
odd	even	0	1	$\frac{d_1-1}{2}$	$\frac{d_2-2}{2}$	$d_1 - 1$	$d_1 - 1$	0	0
odd	odd	0	0	$\frac{d_1-1}{2}$	$\frac{d_2-1}{2}$	0	0	0	0

□

Each of the next 4 subsections provides details for one of the 4 scenarios that are mentioned at the beginning this section. These subsections correspond to scenarios with exactly 1 distinct value, exactly 2 distinct values, exactly 3 distinct values, and exactly 4 distinct values, respectively.

In these next 4 subsections, the set of 3 conditions, that is,

$$\begin{gathered} r_1 s_0 = r_0 s_1, \\ r_0, r_1, s_0, s_1, N \in \mathbb{Z}^+, \text{ and} \\ r_0 + r_1 + s_0 + s_1 = N \end{gathered}$$

are referred to as the *general constraints*. Each of the 4 subsections is associated with a set of constraints. The constraints for a subsection consist of these three general constraints and one or more additional constraints. These additional constraints are given and discussed below.

5.11.1 All four of the values assigned to the variables r_1, s_0, r_0, and s_1 are identical

In this subsection, our goal is to find a systematic way to construct compositions of size 4 for N that satisfy the constraints below and to develop a formula for counting them. A way to help accomplish this is detailed later in this section.

$$\begin{gathered} r_1 s_0 = r_0 s_1 \\ r_0, r_1, s_0, s_1, N \in \mathbb{Z}^+ \\ r_0 + r_1 + s_0 + s_1 = N \\ r_1 = s_0 = r_0 = s_1 \qquad (5.11.1) \end{gathered}$$

Constraint 5.11.1 states that all four of the variables must have the same value.

The lemma in this subsection proves that the cardinality of the set of greatest common divisor pairs for a non-empty set of 4-compositions is either 0 or 1 when all the components of a 4-composition must have the same positive integer value. It also specifies the conditions under which this is true.

Lemma 5.11.2. *Suppose* $G^{(N)}_{1;i_1,i_2,\ldots,i_j}$ *is the set of qualifying* 4*-compositions from the set* $G^{(N)}_{i_1,i_2,\ldots,i_j}$ *that satisfies the general constraints and Constraint 5.11.1 on the previous page, where* $m = \tau(N)$*. and the* j *indices* $i_1, i_2, \ldots,\ i_j$ *range over all the* j*-element subsets of* m *(i.e.,* $1 \le i_1 < i_2 < \cdots < i_j \le m$*). Then*

$$|G^{(N)}_{1;i_1,i_2,\ldots,i_j}| = \begin{cases} 1 & : \quad d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \textit{ is even and } d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \textit{ is even;} \\ 0 & : \quad \textit{otherwise.} \end{cases}$$

Proof. Let $w + x = d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ and $y + z = d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$. Also, let $w, x, y, z \in \mathbb{Z}^+$. For the convenience of the reader, the weak composition mapping function of Definition 5.5.0.6 on page 153 is restated here:

$$\text{dpwcm}(w, x, y, z) \rightarrow (wy, xz, wz, xy).$$

Assume that

$$r_1 = wy,$$
$$s_0 = xz,$$
$$r_0 = wz, \text{ and}$$
$$s_1 = xy.$$

Since the values of r_1, s_0, r_0, and s_1 must be identical, the condition

$$wy = xz = wz = xy$$

must also hold. Inspection of this condition reveals that because $wy = wz$, we can infer that $y = z$. Also, because $xz = wz$, we can infer that $x = w$. These implications mean that we can write

$$\begin{aligned} \mathrm{dpwcm}(w, x, y, z) &= \mathrm{dpwcm}(w, w, y, y) \\ &= (wy, wy, wy, wy). \end{aligned}$$

Each of the four components of the 4-tuple generated by this process clearly has the same value $k = wy$. Furthermore, it is true that both $d^{(N)}_{i_1,i_2,\ldots,i_j}[1] = w + w = 2w$ and that $d^{(N)}_{i_1,i_2,\ldots,i_j}[2] = y + y = 2y$ hold. So, $N = (2w)(2y) = 4k$ and $r_1 = s_0 = r_0 = s_1 = k$ are true. The constraints and the manner in which this proof was constructed show that the only form of N that satisfy these constraints is one in which the value of N is a positive integer that is evenly divisible by 4, the divisor pair components are even positive integers, and $r_1 = s_0 = r_0 = s_1 = N/4$. In this situation, there is exactly one solution. There is no solution in the situation where $N = 4k + m$ with $k \in \mathbb{Z}^+$ and $m \in [3]$. □

5.11.2 Only two of the four values assigned to r_1, s_0, r_0, and s_1 are mutually distinct

The goal, in this case, is to find a systematic way to construct compositions of size 4 for N that satisfy the constraints below and to develop a formula for counting them. A way to help accomplish this is detailed later in this section.

$$\begin{gathered} r_1 s_0 = r_0 s_1 \qquad (5.11.2) \\ r_0 + r_1 + s_0 + s_1 = N \\ r_0, r_1, s_0, s_1, N \in \mathbb{Z}^+ \end{gathered}$$

$$r_1 = r_0 \text{ and } s_0 = s_1 \text{ and } r_1 \neq s_1 \tag{5.11.3}$$

$$r_1 = s_1 \text{ and } s_0 = r_0 \text{ and } r_1 \neq s_0 \tag{5.11.4}$$

Collectively, these constraints state that there are only two distinct values (e.g., a and b, with $a \neq b$) among the four values that are assigned to parameters r_1, s_0, r_0, and s_1. Constraint 5.11.3 and Constraint 5.11.4 state that exactly one variable on the left-hand side of Equation 5.11.2 on the previous page has the value a and the other one has the value b. The same statement is true for the right-hand side of this equation.

The upcoming lemma in this subsection proves that the cardinality of the set of greatest common divisor pairs for a non-empty set of 4-compositions is 0, if N is odd, but, otherwise, may be positive or zero. It also specifies the conditions under which the cardinality has a positive value and how to determine this value.

Lemma 5.11.3. *Suppose $G^{(N)}_{2;i_1,i_2,\ldots,i_j}$ is the set of qualifying 4-compositions from the set $G^{(N)}_{i_1,i_2,\ldots,i_j}$ that satisfies the general constraints and either Constraint 5.11.3 or Constraint 5.11.4, where $m = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of m (i.e., $1 \leq i_1 < i_2 < \cdots < i_j \leq m$). Then*

$$|G^{(N)}_{2;i_1,i_2,\ldots,i_j}| = \begin{cases} 2(x\,[d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \textit{ is even}] + y\,[d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \textit{ is even}]), & \textit{if } N \textit{ is even;} \\ 0, & \textit{otherwise;} \end{cases}$$

where $x = \lfloor (d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - 1)/2 \rfloor$, $y = \lfloor (d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - 1)/2 \rfloor$, and the notation [condition] *denotes an expression that evaluates to 1 if* condition *is true but evaluates to 0, otherwise.*

Proof. Let $w + x = d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ and $y + z = d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$. Also, let $w, x, y, z \in \mathbb{Z}^+$. For the convenience of the reader, the weak composition mapping function of Definition 5.5.0.6 on page 153 is restated here:

$$\mathrm{dpwcm}(w, x, y, z) \rightarrow (wy, xz, wz, xy).$$

Assume that

$$
\begin{aligned}
&r_1 = wy,\\
&s_0 = xz,\\
&r_0 = wz, \text{ and}\\
&s_1 = xy.
\end{aligned}
$$

Case 1: The values of r_1 and r_0 must be identical, the values of s_0 and s_1 must be identical, and $r_1 \neq s_0$.

The first 2 of these 3 conditions mean that the conditions

$$wy = wz$$

and

$$xz = xy$$

must also hold. From these last 2 conditions, we can infer that $y = z$. This inferred condition, plus the $r_1 \neq s_0$ condition, means that the conditions

$$(wy \neq xz) \text{ and } (y = z) \text{ yield the condition } wz \neq xz.$$

From this we can infer that $w \neq x$. These implications mean that we can write

$$
\begin{aligned}
\text{dpwcm}(w, x, y, z) &= \text{dpwcm}(w, x, y, y)\\
&= (wy, xy, wy, xy). \qquad (5.11.5)
\end{aligned}
$$

It can be readily seen from the result of the mapping for Equation 5.11.5 that, because

w is not equal to x, then the 4-tuple has exactly two mutually distinct values (i.e., wy and xy) among its four components. Now, the expression

$$N = (w + x)(y + z)$$

can be rewritten as

$$N = (w + x)(y + y) = (w + x)(2y)$$

because we established earlier that the condition $y = z$ holds. Since w and x must have different values, the number a of qualifying strong 4-compositions for the

$$d^{(N)}_{i_1,i_2,\ldots,i_j}[1] = w + x$$

part of this case is twice the number of distinct 2-partitions of $d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ because each (w, x) pair contributes two permutations to the factor a for this side of the total count. The number b of qualifying strong 4-compositions for the

$$d^{(N)}_{i_1,i_2,\ldots,i_j}[2] = 2y$$

part of this case is either 1 or 0. It is 1 if $d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$ is an even number; otherwise, it is 0. The total number of qualifying strong 4-compositions for this case is the product of a and b. Note that this product is 0 if $d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$ is an odd number.

Case 2: The values of r_1 and s_1 must be identical, the values of s_0 and r_0 must be identical, and $r_1 \neq s_0$.

The analysis for this case follows a similar pattern to the one for Case 1. The first 2 of the 3 conditions for this case mean that the conditions

$$wy = xy$$

and

$$xz = wz$$

must also hold. From these last 2 conditions, we can infer that $x = w$. This inferred condition, plus the $r_1 \neq s_0$ condition, means that the conditions

$$(wy \neq xz) \text{ and } (x = w) \text{ yield } xy \neq xz.$$

From this we can infer that $y \neq z$. These implications mean that we can write

$$\begin{aligned} \mathrm{dpwcm}(w, x, y, z) &= \mathrm{dpwcm}(w, w, y, z) \\ &= (wy, wz, wz, wy). \end{aligned} \tag{5.11.6}$$

It can be readily seen from the result of the mapping for Equation 5.11.5 on page 215 that, because y is not equal to z, then the 4-tuple has exactly 2 mutually distinct values (i.e., wy and wz) among its 4 components. Now, the expression

$$N = (w + x)(y + z)$$

can be rewritten as

$$N = (w + x)(y + z) = (2w)(y + z)$$

because we established earlier that the condition $x = w$ holds. Since y and z must have different values, the number a of qualifying strong 4-compositions for the

$$d^{(N)}_{i_1,i_2,\ldots,i_j}[2] = y + z$$

part of this case is twice the number of distinct 2-partitions of $d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$ because each (y, z) pair contributes 2 permutations to the factor b for this side of the total count. The number b of qualifying strong 4-compositions for the

$$d^{(N)}_{i_1,i_2,\ldots,i_j}[1] = 2w$$

part of this case is either 1 or 0. It is 1 if $d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ is an even number; otherwise, it is 0. The total number of qualifying strong 4-compositions for this case is the product of a and b. Note that this product is 0 if $d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ is an odd number. □

5.11.3 Only three of the four values assigned to r_1, s_0, r_0, and s_1 are mutually distinct

The goal in this case is to find a systematic way to construct compositions of size 4 for N that satisfy the constraints below and to develop a formula for counting them. A way to help accomplish this is detailed later in this section.

$$\begin{aligned} r_1 s_0 &= r_0 s_1 \\ r_0 + r_1 + s_0 + s_1 &= N \qquad (5.11.7) \\ r_0, r_1, s_0, s_1, N &\in \mathbb{Z}^+ \\ r_0 = s_1 &= \sqrt{r_1 s_0} \text{ and } r_1 \neq s_0 \qquad (5.11.8) \\ r_1 = s_0 &= \sqrt{r_0 s_1} \text{ and } r_0 \neq s_1 \qquad (5.11.9) \end{aligned}$$

Constraint 5.11.8 and Constraint 5.11.9 state that 2 of the variables must have the same value (e.g., c) and that the other 2 variables must have values that are different from each other and that are also different from c. A further requirement is that the values for both $\sqrt{r_1 s_0}$ and $\sqrt{r_0 s_1}$ must be members of $\mathbb{Z}^+$.

The lemma in this subsection proves that the cardinality of the set of greatest common divisor pairs for a non-empty set of 4-compositions is always an integral multiple of 4. It also specifies how to determine this value.

Lemma 5.11.4. *Suppose $G^{(N)}_{3;i_1,i_2,\ldots,i_j}$ is the set of qualifying 4-compositions from the set $G^{(N)}_{i_1,i_2,\ldots,i_j}$ that satisfies the general constraints and either Constraint 5.11.8 or Constraint 5.11.9, where $m = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of m (i.e., $1 \le i_1 < i_2 < \cdots < i_j \le m$). Then $|G^{(N)}_{3;i_1,i_2,\ldots,i_j}| = 4 \times \lfloor (\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2 \rfloor$.*

Proof. Let $w + x = d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ and $y + z = d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$. Also, let $w, x, y, z \in \mathbb{Z}^+$. For the convenience of the reader, the weak composition mapping function of Definition 5.5.0.6 on page 153 is restated here:

$$\text{dpwcm}(w, x, y, z) \rightarrow (wy, xz, wz, xy).$$

Assume that

$$\begin{aligned} r_1 &= wy, \\ s_0 &= xz, \\ r_0 &= wz, \text{ and} \\ s_1 &= xy. \end{aligned}$$

Case 1: The values of r_0 and s_1 must be equal to the square root of the product of r_1 and s_0, and $r_1 \neq s_0$.

For the discussion below, let $g = \gcd(w + x, y + z)$, the greatest common divisor (GCD) of the sums $w + x$ and $y + z$; $k_1 = (w + x)/g$; and $k_2 = (y + z)/g$.

This means that, in this context, $r_1 s_0 = r_0 s_1$ is equivalent to $r_1 s_0 = (r_0)^2$ and that $r_1 + s_0 + 2r_0 = N$. If we assume that $g = e + f$, we can rewrite Equation 5.11.7 on page 218 as

$$\begin{aligned} N &= k_1(e + f)k_2(e + f) \\ &= k_1 k_2 (e + f)^2 \\ &= k_1 k_2 (e^2 + f^2 + 2ef) \\ &= r_1 + s_0 + 2r_0. \end{aligned}$$

From this rewrite, we can state that solutions can be obtained by making assignments of the form shown in these sets:

$$\{r_1 \leftarrow k_1 k_2 e^2, s_0 \leftarrow k_1 k_2 f^2, r_0 \leftarrow s_1 \leftarrow k_1 k_2 ef\}$$

or

$$\{r_1 \leftarrow k_1 k_2 f^2, s_0 \leftarrow k_1 k_2 e^2, r_0 \leftarrow s_1 \leftarrow k_1 k_2 ef\}.$$

In order for these sets of assignments to satisfy Constraint 5.11.8 on page 218, it is required that the value of e must be different than the value for f. Without this requirement, there would only be 1 distinct value being assigned in these sets of assignments, thereby violating the constraint that there must be 3 distinct values. The manner in which we calculate these values ensure that this constraint is met and can be proved rather easily. Without loss of generality, assume that the value of e is less than the value of f. Therefore, the values represented by e^2 and f^2 must be mutually different. Lastly, because e and f represent different values, it is also true that $e^2 \neq ef$ and that $f^2 \neq ef$.

From this we can see that the values for r_1, s_0, and r_0 are mutually different with the values for r_0 and s_1 being equal to each other.

The number of value pairs that satisfy this requirement is the same as the number of distinct 2-partitions of the sum $e+f$, that is, $\lfloor(\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2\rfloor$. Since each distinct 2-partition of $e+f$ has 2 representatives in the set $\{(v_1, v_2)|v_1 + v_2 = e+f, v_1 \neq v_2, v_1 \in \mathbb{Z}^+\}$, due to symmetry, the overall contribution for this case is $2 \times \lfloor(\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2\rfloor$.

Case 2: The values of r_1 and s_0 must be equal to the square root of the product of r_1 and s_0 , and $r_0 \neq s_1$.

For the discussion below, let $g = \gcd(w+x, y+z)$, the GCD of the sums $w+x$ and $y+z$; $k_1 = (w+x)/g$; and $k_2 = (y+z)/g$.

The analysis for this case is similar to that for the prior case. The above condition means that, in this context, $r_1 s_0 = r_0 s_1$ is equivalent to $(r_1)^2 = r_0 s_1$ and that $2r_1 + r_0 + s_1 = N$. If we assume that $g = e+f$, we can rewrite Equation 5.11.7 on page 218 as

$$\begin{aligned} N &= k_1(e+f)k_2(e+f) \\ &= k_1k_2(e+f)^2 \\ &= k_1k_2(e^2+f^2+2ef) \\ &= 2r_1 + r_0 + s_1. \end{aligned}$$

From this, we can state that solutions can be obtained by making assignments of the form shown in these sets:

$$\{r_1 \leftarrow s_0 \leftarrow k_1k_2ef, r_0 \leftarrow k_1k_2e^2, s_1 \leftarrow k_1k_2f^2\}$$

or

$$\{r_1 \leftarrow s_0 \leftarrow k_1k_2ef, r_0 \leftarrow k_1k_2f^2, s_1 \leftarrow k_1k_2e^2\}.$$

In order for these sets of assignments to satisfy Constraint 5.11.9 on page 218, it is required that the value of e must be different than the value for f. Without this requirement, there would only be 1 distinct value being assigned in these sets of assignments, thereby, violating the constraint that there must be 3 distinct values. The manner in which we calculate these values ensures that this constraint is met and can be proved rather easily. Without loss of generality, assume that the value of e is less than the value of f. Therefore, the values represented by e^2 and f^2 must be mutually different. Lastly, because e and f represent different values, it is also true that $e^2 \neq ef$ and that $f^2 \neq ef$. From this we can see that the values for r_1, r_0, and s_1 are mutually different with r_1 and s_0 being equal to each other.

The number of value pairs that satisfy this requirement is the same as the number of distinct 2-partitions of the sum $e+f$, that is, $\lfloor(\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2])-1)/2\rfloor$. Since each distinct 2-partition of $e+f$ has two representatives in the set $\{(v_1, v_2)|v_1 + v_2 = e+f, v_1 \neq v_2, v_1 \in \mathbb{Z}^+\}$, due to commutativity, the overall contribution for this case is $2 \times \lfloor(\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2\rfloor$.

Once we combine the results for the two cases, we find that the total number of qualifying 4-compositions is $4 \times \lfloor(\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2\rfloor$. □

5.11.4 All four of the values assigned to r_1, s_0, r_0, and s_1 are mutually distinct

Our goal in this case is similar to that in the immediately prior case: find a systematic way to construct compositions of size 4 for N that satisfy the constraints below and to develop a formula for counting them. The main difference between this case and its immediate predecessor is that the values associated with the four variables r_1, s_0, r_0, and

s_1 must be mutually distinct.

$$\begin{aligned} r_1 s_0 &= r_0 s_1 \\ r_0 + r_1 + s_0 + s_1 &= N \\ r_0, r_1, s_0, s_1, N &\in \mathbb{Z}^+ \\ r_0, r_1, s_0, \text{ and } s_1 &\text{ have mutually distinct values} \end{aligned} \tag{5.11.10}$$

The lemmas in this subsection provide a mechanism to calculate the cardinality of the set of greatest common divisor pairs for a non-empty set of 4-compositions. The cardinality calculations they describe are considerably more complex that those discussed in the previous three subsections. A large portion of this complexity is due to the fact that the component values must be pairwise distinct.

Lemma 5.11.5. *Suppose $G^{(N)}_{4;i_1,i_2,\ldots,i_j}$ is the set of qualifying 4-compositions from the set $G^{(N)}_{i_1,i_2,\ldots,i_j}$ that satisfies the general constraints and Constraint 5.11.10, where $m = \tau(N)$ and the j indices $i_1, i_2, \ldots, i_j$ range over all the j-element subsets of m (i.e., $1 \le i_1 < i_2 < \cdots < i_j \le m$). Then*

$$|G^{(N)}_{4;i_1,i_2,\ldots,i_j}| = |G'^{(N)}_{4;i_1,i_2,\ldots,i_j}| - 4\lfloor(\gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]) - 1)/2\rfloor$$

where

$$\begin{aligned} |G'^{(N)}_{4;i_1,i_2,\ldots,i_j}| &= (d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - 1 - [d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \textit{ is even}]) \times \\ &\quad (d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - 1 - [d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \textit{ is even}]). \end{aligned}$$

Proof. Every member of $G^{(N)}_{i_1,i_2,\ldots,i_j}$ is of the form (wy, xz, wz, xy), where (w, x, y, x) is a member of $D^{(N)}_{i_1,i_2,\ldots,i_j}$. Those members of $G^{(N)}_{i_1,i_2,\ldots,i_j}$ that also qualify for membership in $G^{(N)}_{4;i_1,i_2,\ldots,i_j}$ are those members that have mutually distinct component values. The number of these members can be determined by first noticing that 4-tuples from $D^{(N)}_{i_1,i_2,\ldots,i_j}$, where the values of either the first 2 components, the last 2 components, or all 4 components are the same, cannot possibly be a member of $G^{(N)}_{4;i_1,i_2,\ldots,i_j}$ because the dpwcm function, when applied to such a tuple, produces a 4-composition that has the same value for two or more of its components. Members of $D^{(N)}_{i_1,i_2,\ldots,i_j}$ that have this characteristic (i.e., $d^{(N)}_{i_1,i_2,\ldots,i_j}[1]$ is even, $d^{(N)}_{i_1,i_2,\ldots,i_j}[2]$ is even, or both are even) cannot be members of $G^{(N)}_{i_1,i_2,\ldots,i_j}$ because at least one component of the divisor pair $d^{(N)}_{i_1,i_2,\ldots,i_j}$ has an ordered sum of the form $a+a$ associated with it. This ordered sum form cannot occur for a component when the value of that component is odd. Let

$$G'^{(N)}_{4;i_1,i_2,\ldots,i_j} = \{\text{dpwcm}(w,x,y,z) | (w,x,y,z) \in D^{(N)}_{i_1,i_2,\ldots,i_j}, w \neq x, \text{ and } y \neq z\}.$$

correspond to the members of $D^{(N)}_{i_1,i_2,\ldots,i_j}$ that do not have associated ordered sums of the form $a + a$.

The cardinality of this set, denoted $|G'^{(N)}_{4;i_1,i_2,\ldots,i_j}|$, is

$$(d^{(N)}_{i_1,i_2,\ldots,i_j}[1] - 1 - [d^{(N)}_{i_1,i_2,\ldots,i_j}[1] \text{ is even}])(d^{(N)}_{i_1,i_2,\ldots,i_j}[2] - 1 - [d^{(N)}_{i_1,i_2,\ldots,i_j}[2] \text{ is even}])$$

and can be viewed as an approximation to the cardinality of $G^{(N)}_{4;i_1,i_2,\ldots,i_j}$.

All the members of set $G'^{(N)}_{4;i_1,i_2,\ldots,i_j}$, except for those that have the same value for two or more of its components, are also members of $G^{(N)}_{4;i_1,i_2,\ldots,i_j}$. If we can determine how to count those exceptions, then we can obtain the cardinality of $G^{(N)}_{4;i_1,i_2,\ldots,i_j}$ by subtracting the number of these exceptions from the cardinality of set $G'^{(N)}_{4;i_1,i_2,\ldots,i_j}$. The following paragraphs discuss how to derive an expression for the number of exceptions.

We start by noting that any member of $G'^{(N)}_{4;i_1,i_2,\ldots,i_j}$, that is an exception, has exactly two components with the same value. These members are of the form (c, c, d, e) or (c, d, e, e) with the values for c, d, and e being mutually distinct.

When the situation exists where $wy = xz$ is true, the ratio of w to x is the same as the ratio of z to y. If we let $g = a + b = \gcd(w + x, y + z) = \gcd(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2])$; $k_1 = d^{(N)}_{i_1,i_2,\ldots,i_j}[1]/g$; and $k_2 = d^{(N)}_{i_1,i_2,\ldots,i_j}[2]/g$, we can rewrite

$$(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2]))$$

as

$$(k_1(a + b), k_2(a + b)).$$

From this, we see that $k_1a : k_1b$ is equivalent to $k_2a : k_2b$. Now, let $w = k_1a$, $x = k_1b$, $y = k_2b$, and $z = k_2a$. This allows us to write

$$\begin{aligned}
\mathrm{dpwcm}(w, x, y, z) &= (wy, xz, wz, xy) \\
&= (k_1ak_2b, k_1bk_2a, k_1ak_2a, k_1bk_2b) \\
&= (k_1k_2ab, k_1k_2ba, k_1k_2aa, k_1k_2bb) \\
&= (k_1k_2ab, k_1k_2ab, k_1k_2a^2, k_1k_2b^2).
\end{aligned}$$

When the situation exists where $wz = xy$ is true, the ratio of w to x is the same as the ratio of y to z. Like before, we can rewrite

$$(d^{(N)}_{i_1,i_2,\ldots,i_j}[1], d^{(N)}_{i_1,i_2,\ldots,i_j}[2])$$

as

$$(k_1(a + b), k_2(a + b)).$$

From this, we see that $k_1a : k_1b$ is equivalent to $k_2a : k_2b$. Now, let $w = k_1a$, $x = k_1b$,

$y = k_2a$, and $z = k_2b$. This allows us to write

$$\begin{aligned}\text{dpwcm}(w, x, y, z) &= (wy, xz, wz, xy)\\ &= (k_1ak_2a, k_1bk_2b, k_1ak_2b, k_1bk_2a)\\ &= (k_1k_2aa, k_1k_2bb, k_1k_2ab, k_1k_2ba)\\ &= (k_1k_2a^2, k_1k_2b^2, k_1k_2ab, k_1k_2ab).\end{aligned}$$

Each situation discussed above has two possibilities associated with it. The complete collection of four possibilities for a pair of unequal a and b values is enumerated in Table 5.8. Since the a and b values used in that table are assumed to be unequal, there are only three distinct values per row. Therefore, $G'^{(N)}_{4;i_1,i_2,\ldots,i_j}$ has $4\lfloor(g-1)/2\rfloor$ members that are not eligible to be members of $G^{(N)}_{4;i_1,i_2,\ldots,i_j}$. Putting this all together, we obtain

$$|G^{(N)}_{4;i_1,i_2,\ldots,i_j}| = |G'^{(N)}_{4;i_1,i_2,\ldots,i_j}| - 4\lfloor(g-1)/2\rfloor.$$

□

Table 5.8: The Four Possibilities for Two Duplicate Components.

situation	w	x	y	z	wy	xz	wz	xy
$(w : x \equiv z : y) \Rightarrow wy = xz$	k_1a	k_1b	k_2b	k_2a	k_1k_2ab	k_1k_2ab	$k_1k_2a^2$	$k_1k_2b^2$
$(w : x \equiv z : y) \Rightarrow wy = xz$	k_1b	k_1a	k_2a	k_2b	k_1k_2ab	k_1k_2ab	$k_1k_2b^2$	$k_1k_2a^2$
$(w : x \equiv y : z) \Rightarrow wz = xy$	k_1a	k_1b	k_2a	k_2b	$k_1k_2a^2$	$k_1k_2b^2$	k_1k_2ab	k_1k_2ab
$(w : x \equiv y : z) \Rightarrow wz = xy$	k_1b	k_1a	k_2b	k_2a	$k_1k_2b^2$	$k_1k_2a^2$	k_1k_2ab	k_1k_2ab

Before proceeding further, we briefly recap what we have established with the last four lemmas. These lemmas (i.e., Lemma 5.11.2 on page 212, Lemma 5.11.3 on page 214, Lemma 5.11.4 on page 219, and Lemma 5.11.5 on page 223) developed expressions for counting the number of events in Quadrant IV when $p' = t'$ holds and each of the parameters r_1, r_0, s_1, and s_0 have positive integer values. What we are mainly interested in, though, is the total number of events in Quadrant IV when $p' > t'$ holds and the parameters have positive values.

By Lemma 5.2.1 on page 146, we know that, if we can figure out the count for the former number of events (i.e., the number of events in Quadrant IV when $p' = t'$ holds and each of the parameters r_1, r_0, s_1, and s_0 have positive integer values), then can use that value to obtain the one for the latter number of events (i.e., the total number of events in Quadrant IV when $p' > t'$ holds and the parameters have positive values). We can use of one of the several equivalent forms of the Principle of Inclusion-Exclusion to do that.

Lemma 5.11.1 on page 206 uses this principle to establish a formula for the count of the former number of events (i.e., the number of events in Quadrant IV when $p' = t'$ holds and each of the parameters r_1, r_0, s_1, and s_0 have positive integer values). Later, we use this result, and the result from Lemma 5.11.6, to count the latter number of events.

Lemma 5.11.6.

$$\textit{The contribution is} \begin{cases} \dfrac{C_4(N) - \left|\bigcup_{1\leq i\leq m} G_i^{(N)}\right|}{2}, & \textit{if } N \geq 1; \\ 0, & \textit{otherwise;} \end{cases}$$

when the conditions $p' > t'$, $p \in (0,1)$, and $q \in (0,1)$ are all true.

Proof. The expression

$$\left|\bigcup_{1\leq i\leq m} G_i^{(N)}\right|$$

calculates the number of 4-compositions of $N \geq 0$ where $r_1 s_0 = r_0 s_1$. By Lemma 5.2.1 on page 146, the number of 4-compositions of N that satisfy $r_1 s_0 > r_0 s_1$ is the same as the number of 4-compositions of N that satisfy $r_1 s_0 < r_0 s_1$. Therefore, the number of

4-compositions of N that satisfy $p' > t'$ is

$$\frac{C_4(N) - \left|\bigcup_{1 \leq i \leq m} G_i^{(N)}\right|}{2}.$$

□

After putting all of this together, we obtain

$$\mathcal{Q}'_{\text{CLM}} = \Pr(p' > t') = \begin{cases} \frac{2\binom{N-1}{2} + N - 1 + \frac{C_4(N) - \left|\bigcup_{1 \leq i \leq m} G_i^{(N)}\right|}{2}}{\binom{N+3}{3}}, & \text{if } N \geq 1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.11.11}$$

5.12 Mean and Variance

Definition 5.12.0.1. If X is a discrete random variable with probability distribution $f(x)$, then the expected value (i.e., mean) and variance of X are

$$E[X] = \sum_x x f(x)$$

and

$$Var[X] = \sum_x (x - E[X])^2 = E[X^2] - (E[X])^2, \text{ respectively.}$$

Walpole (2002); Blumenfeld (2001); Mood et al. (1973).

Definition 5.12.0.2. Let BooleToNat(x), the Boolean-to-natural-number transformation function, be defined, as follows, for any Boolean-valued expression x.

$$\text{BooleToNat}(x) = \begin{cases} 1, & \text{if } x \text{ is } \textit{true;} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 5.12.0.3. Given a sample space Ω, the discrete random variable $\mathcal{B}$ is a function such that, for each outcome $\omega \in \Omega$,

$$\mathcal{B}(\omega) = \text{BooleToNat}(p'_\omega > t'_\omega),$$

where Ω is the set of weak 4-compositions of N, and p'_ω and q'_ω are the p' and q' values, respectively, for this ω.

The last two definitions can be used to rewrite $\mathcal{Q}'_{\text{CLM}}$ to show that $\mathcal{Q}'_{\text{CLM}}$ is simply the expected value, or mean, of $\mathcal{B}$.

Lemma 5.12.1. *The expected value of* $\mathcal{B}$ *is* $\mathcal{Q}'_{\text{CLM}}$. *That is,*

$$\begin{aligned} E[\mathcal{B}] &= \sum_{\omega \in \Omega} \mathcal{B}(\omega) \binom{N+4-1}{4-1}^{-1} \\ &= \frac{\sum_{\omega \in \Omega} \mathcal{B}(\omega)}{\binom{N+3}{3}} \\ &= \mathcal{Q}'_{\text{CLM}}. \end{aligned}$$

Proof. The number of weak 4-compositions for N is $\binom{N+4-1}{4-1}$, by Equation 2.2.2 on page 26. These compositions are equally likely, each with a probability of $\binom{N+4-1}{4-1}^{-1}$. The random variable $\mathcal{B}$ is binary-valued, with values that are either 0 or 1; i.e., $\mathcal{B}(\omega) \in \{0, 1\}$. Each weak 4-composition of N, denoted by ω, where the p' value is greater than the corresponding t' value, is associated with a $\mathcal{B}(\omega)$ value of 1; otherwise, the $\mathcal{B}(\omega)$ value is 0. □

Lemma 5.12.2. *The variance of* $\mathcal{B}$ *is*

$$Var[\mathcal{B}] = E[\mathcal{B}^2] - (E[\mathcal{B}])^2 \tag{5.12.1}$$

$$= E[\mathcal{B}] - (E[\mathcal{B}])^2 \tag{5.12.2}$$

$$= \mathcal{Q}'_{\text{CLM}} - (\mathcal{Q}'_{\text{CLM}})^2 \tag{5.12.3}$$

$$= \mathcal{Q}'_{\text{CLM}}(1 - \mathcal{Q}'_{\text{CLM}}). \tag{5.12.4}$$

Proof. Line 5.12.1 follows from Definition 5.12.0.1 on page 228. The value of $\mathcal{B}(\omega)$ is always the same as that of $(\mathcal{B}(\omega))^2$ because, by Definition 5.12.0.3 on page 228, $\mathcal{B}(\omega)$ can only take on the values 0 and 1. This is the justification for Line 5.12.2. The justification for Line 5.12.3 comes from Lemma 5.12.1 on the preceding page because it established that $\mathcal{Q}'_{\text{CLM}}$ is the expected value of $\mathcal{B}$, that is, $\mathcal{Q}'_{\text{CLM}} = E[\mathcal{B}]$. The expression in Line 5.12.4 is a basic factoring of the expression in Line 5.12.3. □

5.13 Example: An Application of the Principle of Inclusion-Exclusion

Many concepts, definitions, and lemmas have been specified in the previous sections. Some of them may have been harder, or easier, to grasp than others. The purpose of this example is to enhance the readers' understanding of these entities. To keep this example manageable, from the perspective of combinatorial explosion avoidance, it is assumed that the document collection only has twelve documents (i.e., $N = 12$).

By the Fundamental Theorem of Arithmetic, $N = 2^2 \cdot 3^1$. Therefore, N has $(2 + 1)(1 + 2) = 6$ distinct positive integer divisors. Likewise, N also has 6 divisor pairs. These pairs are $(1, 12)$, $(2, 6)$, $(3, 4)$, $(4, 3)$, $(6, 2)$, and $(12, 1)$; their indexes range from 1 to 6, respectively (e.g., the index of divisor pair $(1, 12)$ is 1, that of divisor pair $(2, 6)$ is 2, and the index of divisor pair $(12, 1)$ is 6). These divisor pair mappings appear in Table 5.1 on page 152.

The divisor pairs and their associated D and G sets are listed in Table 5.3 on page 160. Note that neither divisor pair $(1, 12)$ nor divisor pair $(12, 1)$ appear in this table because each has at least one component that cannot be expressed as a sum of two positive integers. By Definition 5.5.0.4 on page 153 and Definition 5.5.0.5 on page 153, each component must be expressible as a sum of two positive integers. Due to this restriction, the G set for divisor pair $(1, 12)$ is the empty set. For the same reason, the G set for divisor pair $(12, 1)$ is also the empty set.

The implication of the information in the previous paragraph is that the problem of determining the number of qualifying 4-compositions for $N = 12$ can be reduced to the problem of finding the cardinality of the union of the G sets for just these four divisor pairs: $(2, 6)$, $(3, 4)$, $(4, 3)$, and $(6, 2)$. The Principle of Inclusion-Exclusion (POIE) is a general purpose combinatorial technique that can be used to determine this cardinality. Its use is illustrated in the discussion that constitutes the remainder of this section.

Assume that there are n sets to be unioned. The POIE works by first computing the sum of the cardinalities of all the 1-subsets, then subtracting the sum of the cardinalities of the intersection of all the 2-subsets, then adding the sum of the cardinalities of the $3-$subset intersection, and so on. This alternation between addition and subtraction continues up to and including the determination of the the cardinality of the intersection of all n sets.

5.13.1 The 1-subsets and Their Cardinalities

The 1-subsets were obtained from Table 5.3 on page 160 and are made explicit below for the convenience of the reader.

$$\begin{aligned}\tilde{G}_1^{(8)} = \{ & \\ & (0,8,0,0),(0,7,0,1),(0,6,0,2),(0,5,0,3),(0,4,0,4),(0,3,0,5),\\ & (0,2,0,6),(0,1,0,7),(0,0,0,8),(0,0,8,0),(1,0,7,0),(2,0,6,0),\\ & (3,0,5,0),(4,0,4,0),(5,0,3,0),(6,0,2,0),(7,0,1,0),(8,0,0,0)\\ \}. &\end{aligned}$$

$$\begin{aligned}\tilde{G}_2^{(8)} = \{ & \\ & (0,8,0,0),(0,6,0,2),(0,4,0,4),(0,2,0,6),(0,0,0,8),(0,4,4,0),\\ & (1,3,3,1),(2,2,2,2),(3,1,1,3),(4,0,0,4),(0,0,8,0),(2,0,6,0),\\ & (4,0,4,0),(6,0,2,0),(8,0,0,0)\\ \}. &\end{aligned}$$

$$\begin{aligned}\tilde{G}_3^{(8)} = \{ & \\ & (0,8,0,0),(0,4,0,4),(0,0,0,8),(0,6,2,0),(1,3,1,3),(2,0,0,6),\\ & (0,4,4,0),(2,2,2,2),(4,0,0,4),(0,2,6,0),(3,1,3,1),(6,0,0,2),\\ & (0,0,8,0),(4,0,4,0),(8,0,0,0)\\ \}. &\end{aligned}$$

$$\begin{aligned}\tilde{G}_4^{(8)} = \{ & \\ & (0,8,0,0),(0,0,0,8),(0,7,1,0),(1,0,0,7),(0,6,2,0),(2,0,0,6),\\ & (0,5,3,0),(3,0,0,5),(0,4,4,0),(4,0,0,4),(0,3,5,0),(5,0,0,3),\\ & (0,2,6,0),(6,0,0,2),(0,1,7,0),(7,0,0,1),(0,0,8,0),(8,0,0,0)\\ \}. &\end{aligned}$$

The sum of their cardinalities is $18 + 15 + 15 + 18 = 66$.

5.13.2 The 2-subsets and Their Cardinalities

The 2-subset intersections are listed below.

$$\tilde{G}_1^{(8)} \cap \tilde{G}_2^{(8)} = \{$$

$$(0,0,0,8),(0,0,8,0),(0,2,0,6),(0,4,0,4),(0,6,0,2),(0,8,0,0),$$

$$(2,0,6,0),(4,0,4,0),(6,0,2,0),(8,0,0,0)$$

$$\}.$$

$$\tilde{G}_1^{(8)} \cap \tilde{G}_3^{(8)} = \{$$

$$(0,0,0,8),(0,0,8,0),(0,4,0,4),(0,8,0,0),(4,0,4,0),(8,0,0,0)$$

$$\}.$$

$$\tilde{G}_1^{(8)} \cap \tilde{G}_4^{(8)} = \{$$

$$(0,0,0,8),(0,0,8,0),(0,8,0,0),(8,0,0,0)$$

$$\}.$$

$$\tilde{G}_2^{(8)} \cap \tilde{G}_3^{(8)} = \{$$

$$(0,0,0,8),(0,0,8,0),(0,4,0,4),(0,4,4,0),(0,8,0,0),(2,2,2,2),$$

$$(4,0,0,4),(4,0,4,0),(8,0,0,0)$$

$$\}.$$

$$\tilde{G}_2^{(8)} \cap \tilde{G}_4^{(8)} = \{$$

$$(0,0,0,8),(0,0,8,0),(0,4,4,0),(0,8,0,0),(4,0,0,4),(8,0,0,0)$$

$$\}.$$

$$\tilde{G}_3^{(8)} \cap \tilde{G}_4^{(8)} = \{ (0,0,0,8),(0,0,8,0),(0,2,6,0),(0,4,4,0),(0,6,2,0),(0,8,0,0), (2,0,0,6),(4,0,0,4),(6,0,0,2),(8,0,0,0) \}.$$

The sum of their cardinalities is $10 + 6 + 4 + 9 + 6 + 10 = 45$.

5.13.3 The 3-subsets and Their Cardinalities

The 3-subset intersections are listed below.

$$\tilde{G}_1^{(8)} \cap \tilde{G}_2^{(8)} \cap \tilde{G}_3^{(8)} = \{(0,0,0,8),(0,0,8,0),(0,4,0,4),(0,8,0,0),(4,0,4,0),(8,0,0,0)\}.$$
$$\tilde{G}_1^{(8)} \cap \tilde{G}_2^{(8)} \cap \tilde{G}_4^{(8)} = \{(0,0,0,8),(0,0,8,0),(0,8,0,0),(8,0,0,0)\}.$$
$$\tilde{G}_1^{(8)} \cap \tilde{G}_3^{(8)} \cap \tilde{G}_4^{(8)} = \{(0,0,0,8),(0,0,8,0),(0,8,0,0),(8,0,0,0)\}.$$
$$\tilde{G}_2^{(8)} \cap \tilde{G}_3^{(8)} \cap \tilde{G}_4^{(8)} = \{(0,0,0,8),(0,0,8,0),(0,4,4,0),(0,8,0,0),(4,0,0,4),(8,0,0,0)\}.$$

The sum of their cardinalities is $6 + 4 + 4 + 6 = 20$.

5.13.4 The 4-subset and Its Cardinality

The 4-subset intersection is

$$\tilde{G}_1^{(8)} \cap \tilde{G}_2^{(8)} \cap \tilde{G}_3^{(8)} \cap \tilde{G}_4^{(8)} = \{(0,0,0,8),(0,0,8,0),(0,8,0,0),(8,0,0,0)\}.$$

The cardinality is 4.

5.13.5 The Resultant Cardinality

By the POIE, the resultant cardinality for these unioned sets is $66 - 45 + 20 - 4 = 37$. This can be verified rather easily by noticing that, of these 37 distinct generally qualifying weak 4-compositions, several occur multiple times among the members of the $\tilde{G}$ sets.

These 4 members occur four times each:

$$(0,0,0,8),(0,0,8,0),(0,8,0,0), \text{ and } (8,0,0,0).$$

These 4 members occur three times each:

$$(0,4,0,4),(0,4,4,0),(4,0,0,4), \text{ and } (4,0,4,0).$$

These 9 members each occur twice:

$$(0,2,0,6),(0,2,6,0),(0,6,2,0),(2,0,0,6),(2,0,6,0),$$
$$(2,2,2,2),(0,6,0,2),(6,0,0,2), \text{ and } (6,0,2,0).$$

The effect of this on the count for the union is that the first sum(i.e., 65) in the expression for the resultant cardinality of the 1-subsets is an overcount because these 17 members are counted multiple times. In general, the first sum generated by the POIE process is almost always an overcount. Ultimately, this is corrected by a process that alternately subtracts and adds subsequent terms that are associated with the remaining k-subsets where $2 \le k \le$ (the number of 1-subsets).

Table 5.9: Number of Qualifying Contributions-Related Values ($1 \leq N \leq 20$).

N	number of qualifying weak 4-comps	total number of weak 4-comps	$\mathcal{Q}'$ (mean)	$\sqrt{\mathcal{Q}'(1-\mathcal{Q}')}$ (standard deviation)
1	0	4	0	0
2	1	10	0.1	0.3
3	4	20	0.2	0.4
4	9	35	0.257 143	0.437 059
5	18	56	0.321 429	0.467 025
6	28	84	0.333 333	0.471 405
7	46	120	0.383 333	0.486 198
8	64	165	0.387 879	0.487 267
9	90	220	0.409 091	0.491 666
10	119	286	0.416 084	0.492 908
11	160	364	0.439 56	0.496 334
12	195	455	0.428 571	0.494 872
13	254	560	0.453 571	0.497 84
14	306	680	0.45	0.497 494
15	370	816	0.453 431	0.497 827
16	444	969	0.458 204	0.498 25
17	536	1140	0.470 175	0.499 11
18	615	1330	0.462 406	0.498 585
19	732	1540	0.475 325	0.499 391
20	829	1771	0.468 097	0.498 981

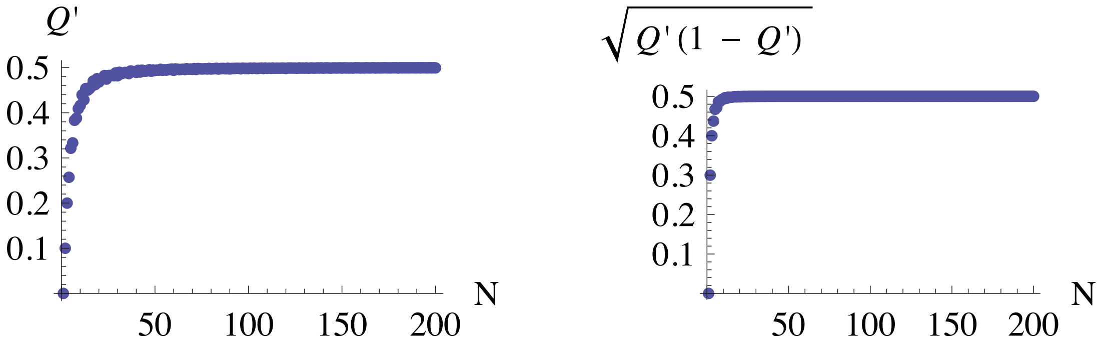


Figure 5.9: Plot of the mean ($\mathcal{Q}'$) and standard deviation $\left(\sqrt{\mathcal{Q}'\,(1-\mathcal{Q}')}\right)$ for $\mathcal{B}$ for the CLM ranking method when $1 \leq N \leq 200$.

5.14 Summary

This chapter presented a combinatoric model of $\mathcal{Q}'$ for the CLM ranking method. It developed counting expressions to determine the cardinality the subset of weak 4-compositions for a document collection of size N. The members of this subset were all the members of the set of weak 4-compositions of N where the values of the members' components satisfied the $p' > t'$ condition.

The chapter started by developing cardinality-counting expressions for Quadrants I, II, and III. Following that, similar expressions were developed for all parts of Quadrant IV, except for the part that corresponded to the joint conditions $p \in (0, 1)$ and $q \in (0, 1)$. Most of the effort for the discussion in this chapter was related to the development of the counting expressions for this exception (as it required special treatment) and to rigorously prove that these expressions were correct.

It was relatively easy to determine the counting expressions for Quadrants I, II, and III, and for all parts of Quadrant IV, except for the part that corresponded to the joint conditions. The reason for this ease was all of the analyses that were discussed in Chapter 4. Since these analyses developed the basic formulas, all that was needed in this chapter was a straightforward combination of these formulas to calculate the desired results. The resultant combinations were closed-form expressions.

By contrast, the development of the counting expressions for the part of Quadrant IV that corresponded to the joint conditions $p \in (0,1)$ and $q \in (0,1)$ was considerably more complex and involved. The expression development was divided into four mutually exclusive cases based on the number of distinct values in each 4-composition. Closed-form expressions were obtained for the first three cases but a closed-form expression was not possible for the fourth case. This last case required the use of the Principle of Inclusion and Exclusion.

Many concepts were developed and introduced for the analyses that occurred in this chapter. Entity-relationship diagrams were used to illustrate the important semantic relationships between these concepts. Near the end of this chapter, a comprehensive example was given, and discussed, to help the reader with the understanding of these concepts and with how the cardinalities were being determined.

Figure 5.9 on the preceding page contains plots of the mean and the standard deviation values for the CLM ranking method. Table 5.9 on page 236 contains the values that were used to create the plots of the first 20 mean and standard deviation values. Research for this chapter indicated that the ordinate (i.e., y-axis) asymptote was 0.5 for both plots. The research also showed that the plotted values for the mean and standard deviation had an overall tendency to increase more than they decreased as the number N of documents in a collection increased from 1 to ∞ (infinity). However, sometimes the mean and standard deviation values temporarily decreased between points N and $N+1$, before the mean and standard deviation values continued their overall increasing trend.

Chapter 6

Combinatoric Models of $\mathcal{Q}'$ for the Inverse Document Frequency and Decision-Theoretic Ranking Methods

The purpose of this chapter is to develop counting expressions that collectively calculate the quality of the inverse document frequency (IDF) and decision-theoretic (DT) ranking methods for a document collection of size N. Some of these expressions come from the general work that took place in Chapter 4. The work in this chapter, along with that in Chapter 5, enable the calculation of the ranking method-specific $\mathcal{Q}'$ values that are referenced in many of the equations that are in Section 7.10 (A Family of ASL Measures), which starts on page 327, and Section 8.2 (The Validation of $\mathcal{Q}'$ Estimates That Were Obtained by Random Sampling), which starts on page 348.

The IDF and DT quality of ranking equations, that are derived later in this chapter, are used in Section 7.8 and Section 7.10 of the next chapter to help develop equations for the normalized and unnormalized search lengths, along with equations for the expected value and variance of these search lengths. These equations also occupy a prominent role in Chapter 8 during the validation of formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ measures.

For the convenience of the reader, we restate the following concepts from Chapter 2 and Section 4.2. From the information retrieval (IR) perspective of this dissertation, these concepts cover the notions of weak composition, composition, and sample space.

A weak composition of size 4 is a collection of N documents where *at least one* of the following conditions is true: the number of relevant documents that contain the query term is 0 (i.e., $r_1 = 0$), the number of relevant documents that do not contain the query term is 0 (i.e., $r_0 = 0$), the number of non-relevant documents that contain the query term is 0 (i.e., $s_1 = 0$), or the number of non-relevant documents that do not contain the query term is 0 (i.e., $s_0 = 0$).

A strong composition of size 4 is a collection of N documents where *all* of the following conditions must be true: the number of relevant documents that contain the query term is positive (i.e., $r_1 \geq 1$), the number of relevant documents that do not contain the query term is positive (i.e., $r_0 \geq 1$), the number of non-relevant documents that contain the query term is positive (i.e., $s_1 \geq 1$), and the number of non-relevant documents that do not contain the query term is positive (i.e., $s_0 \geq 1$).

A sample space for a weak-composition of size k, and N documents, represents all the possible collections of N documents in terms of the k parameters. For example, $k = 4$ in many of the discussions in this chapter and subsequent ones. When $k = 4$, the parameters are r_1, r_0, s_1, and s_0. An *outcome* is an element of this sample space and represents exactly one of its collections.

6.1 Combinatoric Model of $\mathcal{Q}'$ for the IDF Ranking Method

IDF ranking is based on the calculation of a retrieval status value (RSV) that favors terms which are concentrated in only a few documents of a collection. It varies inversely with the number of documents (i.e., $r_1 + s_1$) to which a term is assigned. The typical weight of a document in IDF ranking is

$$\log \frac{N}{n} = -\log \frac{n}{N} = -\log t,$$

where N is the number of documents in the collection and $n = r_1 + s_1$ is the number of documents that contain the query term.

The quality of the IDF ranking method is defined by the following equation:

$$\mathcal{Q}'_{\mathrm{IDF}} = \Pr(p' > t') + \Pr(p' \leq t', t' = \epsilon) \tag{6.1.1}$$

where

$$\epsilon = \begin{cases} N^{-2}, & \text{if } N \geq 2; \\ 10^{-4}, & \text{otherwise.} \end{cases}$$

Since $\mathcal{Q}'_{\mathrm{CLM}} = \Pr(p' > t')$, we can rewrite Equation 6.1.1 as

$$\mathcal{Q}'_{\mathrm{IDF}} = \mathcal{Q}'_{\mathrm{CLM}} + \Pr(p' \leq t', t' = \epsilon). \tag{6.1.2}$$

Our main task in this chapter is to develop an expression that calculates

$$\Pr(p' \leq t', t' = \epsilon).$$

Then, it is a simple matter to combine this expression with the one for $\mathcal{Q}'_{\mathrm{CLM}}$ in order to calculate $\mathcal{Q}'_{\text{IDF}}$.

Table 6.1: Outcomes for the Joint Condition $p' \leq t'$ and $t' = m$.

condition	r_1	r_0	s_1	s_0	p'	t'	m
$N = 0$	0	0	0	0	10^{-4}	10^{-4}	$1 - 10^{-4}$
$N = 1$	0	0	1	0	10^{-4}	$1 - 10^{-4}$	$1 - 10^{-4}$
$N = 1$	1	0	0	0	$1 - 10^{-4}$	$1 - 10^{-4}$	$1 - 10^{-4}$
$N \geq 2$	0	0	N	0	$1/N^2$	$1 - (1/N^2)$	$1 - (1/N^2)$
$N \geq 2$	1	0	$N - 1$	0	$1 - (1/N^2)$	$1 - (1/N^2)$	$1 - (1/N^2)$
$N \geq 2$	2	0	$N - 2$	0	$1 - (1/N^2)$	$1 - (1/N^2)$	$1 - (1/N^2)$
$N \geq 2$	$\cdots$	0	$\cdots$	0	$1 - (1/N^2)$	$1 - (1/N^2)$	$1 - (1/N^2)$
$N \geq 2$	$N - 1$	0	1	0	$1 - (1/N^2)$	$1 - (1/N^2)$	$1 - (1/N^2)$
$N \geq 2$	N	0	0	0	$1 - (1/N^2)$	$1 - (1/N^2)$	$1 - (1/N^2)$

Table 6.1 enumerates the possible outcomes for the contribution count associated with

$$\Pr(p' \leq t', t' = \epsilon).$$

One can readily see by inspection that the contribution count is $N + 1$ for $N \geq 0$.

Since there are $\tilde{C}_4(N) = \binom{N+3}{3}$ weak compositions of size 4 in the sample space for an N-document collection, where $N \geq 0$, Equation 6.1.2 on the preceding page can be rewritten as

$$\mathcal{Q}'_{\text{IDF}} = \mathcal{Q}'_{\text{CLM}} + \frac{N+1}{\binom{N+3}{3}}. \tag{6.1.3}$$

6.1.1 Mean and Variance

Definition 6.1.1.1. Given a sample space Ω, the discrete random variable $\mathcal{B}$ is a function such that, for each outcome $\omega \in \Omega$,

$$\mathcal{B}(\omega) = \text{BooleToNat}((p'_\omega > t'_\omega) \text{ or } ((p'_x\omega \leq t'_\omega) \text{ and } (t'_\omega = \epsilon))),$$

where Ω is the set of weak 4-compositions of N, and p'_ω and q'_ω are the p' and q' values, respectively, for this ω.

The last two definitions can be used to rewrite $\mathcal{Q}'_{\mathrm{IDF}}$ to show that $\mathcal{Q}'_{\mathrm{IDF}}$ is simply the expected value, or mean, of $\mathcal{B}$.

Lemma 6.1.1. *The expected value of* $\mathcal{B}$ *is* $\mathcal{Q}'_{\mathrm{IDF}}$. *That is,*

$$\begin{aligned} E[\mathcal{B}] &= \sum_{\omega \in \Omega} \mathcal{B}(\omega) \binom{N+4-1}{4-1}^{-1} \\ &= \frac{\sum_{\omega \in \Omega} \mathcal{B}(\omega)}{\binom{N+3}{3}} \\ &= \mathcal{Q}'_{\mathrm{IDF}}. \end{aligned}$$

Proof. The number of weak 4-compositions for N is $\binom{N+4-1}{4-1}$, by Equation 2.2.2 on page 26. These compositions are equally likely, each with a probability of $\binom{N+4-1}{4-1}^{-1}$. The random variable $\mathcal{B}$ is binary-valued, with values that are either 0 or 1; i.e., $\mathcal{B}(\omega) \in \{0, 1\}$. Each weak 4-composition of N, denoted by ω, where the p' value is greater than the corresponding t' value, has a $\mathcal{B}(\omega)$ value of 1; otherwise, the $\mathcal{B}(\omega)$ value is 0. □

Lemma 6.1.2. *The variance of* $\mathcal{B}$ *is*

$$\begin{aligned} Var[\mathcal{B}] &= E[\mathcal{B}^2] - (E[\mathcal{B}])^2 && (6.1.4) \\ &= E[\mathcal{B}] - (E[\mathcal{B}])^2 && (6.1.5) \\ &= \mathcal{Q}'_{\mathrm{IDF}} - (\mathcal{Q}'_{\mathrm{IDF}})^2 && (6.1.6) \\ &= \mathcal{Q}'_{\mathrm{IDF}}(1 - \mathcal{Q}'_{\mathrm{IDF}}). && (6.1.7) \end{aligned}$$

Proof. Line 6.1.4 follows directly from the concept of variance in Definition 5.12.0.1 on page 228. The value of $\mathcal{B}(\omega)$ is always the same as that of $(\mathcal{B}(\omega))^2$ because, by Definition 6.1.1.1 on the preceding page, $\mathcal{B}(\omega)$ can only take on the values 0 and 1. This is the justification for Line 6.1.5 on the previous page. The justification for Line 6.1.6 on the preceding page comes from Lemma 6.1.1 on the previous page because it established that $\mathcal{Q}'_{\text{IDF}}$ is the expected value of $\mathcal{B}$, that is, $\mathcal{Q}'_{\text{IDF}} = E[\mathcal{B}]$. The expression in Line 6.1.7 on the preceding page is a basic factoring of the expression in Line 6.1.6 on the previous page. □

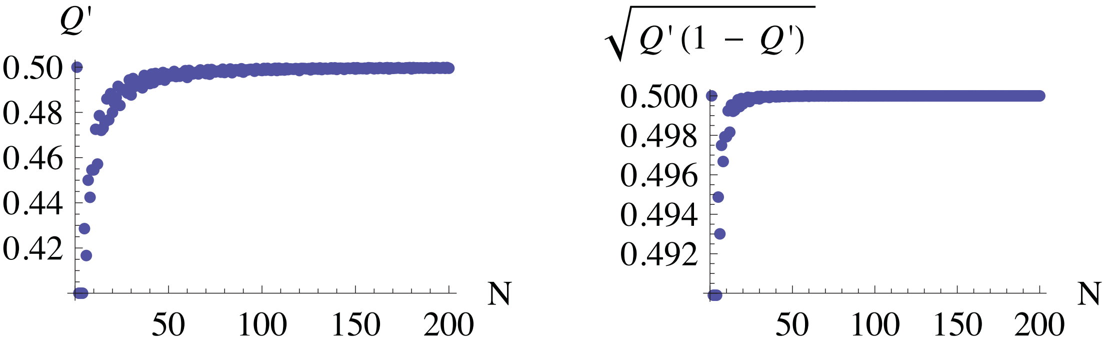


Figure 6.1: Plot of the mean ($\mathcal{Q}'$) and standard deviation $\left(\sqrt{\mathcal{Q}'(1 - \mathcal{Q}')}\right)$ for $\mathcal{B}$ for the IDF ranking method when $1 \leq N \leq 200$.

6.2 Summary for the Inverse Document Frequency Ranking Method

Section 6.1 presents a combinatoric model of $\mathcal{Q}'$ for the IDF ranking method. In this section, a counting expression is developed to determine the cardinality the subset of weak 4-compositions for a document collection of size N. Since the only difference between the $\mathcal{Q}'$ values for the coordination level matching (CLM) and IDF ranking methods is the probability expression

$$\Pr(p' \leq t', t' = \epsilon) \tag{6.2.1}$$

Table 6.2: Number of Qualifying Contributions-Related Values ($1 \leq N \leq 20$).

N	number of qualifying weak 4-comps	total number of weak 4-comps	$\mathcal{Q}'$ (mean)	$\sqrt{\mathcal{Q}'(1-\mathcal{Q}')}$ (standard deviation)
1	2	4	0.5	0.5
2	4	10	0.4	0.489 898
3	8	20	0.4	0.489 898
4	14	35	0.4	0.489 898
5	24	56	0.428 571	0.494 872
6	35	84	0.416 667	0.493 007
7	54	120	0.45	0.497 494
8	73	165	0.442 424	0.496 674
9	100	220	0.454 545	0.497 93
10	130	286	0.454 545	0.497 93
11	172	364	0.472 527	0.499 245
12	208	455	0.457 143	0.498 16
13	268	560	0.478 571	0.499 541
14	321	680	0.472 059	0.499 219
15	386	816	0.473 039	0.499 273
16	461	969	0.475 748	0.499 412
17	554	1140	0.485 965	0.499 803
18	634	1330	0.476 692	0.499 456
19	752	1540	0.488 312	0.499 863
20	850	1771	0.479 955	0.499 598

from Equation 6.1.2 on page 241, the main work that needs to be done is to develop a counting expression for this expression and, then, combine it with the results from the immediately previous chapter to obtain an expression to calculate the quality of ranking measure for the IDF ranking method. The development an expression to calculate Probability Expression 6.2.1 on page 244 is straightforward and results in a closed-form expression.

Figure 6.1 on page 244 contains mean and the standard deviation plots for the IDF ranking method. Table 6.2 on the previous page contains the values that the first 20 points of these plots are based on. Research for this chapter has indicated that the ordinate (i.e., y-axis) asymptote is 0.5 for both plots. The research has also shown that the plotted values for the mean and standard deviation, when $N \geq 2$, have an overall tendency to increase, but sometimes temporarily decreases, between points N and $N+1$ as the number of documents N in a collection increases from 1 to ∞ (infinity).

6.3 A Combinatoric Model of $\mathcal{Q}'$ for the DT Ranking Method

The decision-theoretic ranking discussed in this dissertation is based on binary term independence. This type of term independence assumes that a term is either present or absent in a document and that, if the document has multiple terms, these terms are mutually independent. Multiple occurrences of a term have the same weight as a solitary occurrence of the term. The weight of a document in DT ranking is

$$\log \frac{p/(1-p)}{q/(1-q)}.$$

The equation for $\mathcal{Q}'$ for the decision-theoretic (DT) ranking method is

$$\mathcal{Q}'_{\mathrm{DT}} = \Pr(p' > \max(t', q')) + \Pr(p' \leq \min(t', q')) \tag{6.3.1}$$

where p', t', and q' are defined starting on page 120 in Chapter 4.

We can determine the number of weak 4-compositions in Quadrant IV, where both $p \in (0,1)$ and $q \in (0,1)$ are true, that meet the condition $((p' > \max(t', q'))$ or $(p' \leq \min(t', q')))$ by a strategy that involves breaking the problem into 3 pairwise disjoint cases and counting the number of 4-compositions in each case that meet the condition. By the Law of Trichotomy (Apostol, 1967), if p' and q' are real numbers, then exactly one of these conditions hold: $p' < q'$, $p' = q'$, or $p' > q'$. Each of these conditions corresponds to exactly one of the cases.

Lemma 6.3.1. *Let $x = a/b$; $y = c/d$; $z = (a+c)/(b+d)$; $a, b, c, d \in \mathbb{Z}^+$; and $0 < x, y < 1$. If xRy, then xRz and zRy, where R is either the is-less-than (i.e., $<$), the is-equal-to (i.e., $=$), or the is-greater-than (i.e., $>$) relationship for real numbers.*

Proof. The proof is by cases.

If $x < y$, then $x < z$ and $z < y$.

Rewriting the antecedent provides these initial equivalences:

$$\begin{aligned} x < y &\equiv ad < bc \\ &\equiv a < \frac{bc}{d} \\ &\equiv \frac{ad}{b} < c. \end{aligned} \tag{6.3.2}$$

Now, we rewrite the antecedent to provide the following additional equivalences:

$$x < y \equiv \frac{a}{b} < \frac{c}{d}$$

$$\begin{aligned}
&\equiv \quad \frac{a}{b}\frac{1+\frac{d}{b}}{1+\frac{d}{b}} < \frac{c}{d}\frac{1+\frac{b}{d}}{1+\frac{b}{d}} \\
&\equiv \quad \frac{a+\frac{ad}{b}}{b+d} < \frac{c+\frac{bc}{d}}{b+d}. \qquad (6.3.3)
\end{aligned}$$

Combining Equivalence 6.3.2 on the preceding page, Equivalence 6.3.3, and our assumption that $z = (a+c)/(b+d)$ results in

$$\frac{a+\frac{ad}{b}}{b+d} < \frac{a+c}{b+d} < \frac{c+\frac{bc}{d}}{b+d} \quad \equiv \quad x < z < y,$$

from which we can conclude that, if $x < y$ holds, then $x < z$ and $z < y$ also hold.

If $x = y$, then $x = z$ and $z = y$.

This means that the ratio of the value a to the value b is equivalent to the ratio of the value c to the value d. Since these ratios are equivalent, there exists $k_1, k_2 \in \mathbb{Z}^+$ such that $k_1 = a/g_{\text{ab}} = c/g_{\text{cd}}$ and $k_2 = b/g_{\text{ab}} = d/g_{\text{cd}}$ where $g_{\text{ab}} = \gcd(a,b)$ and $g_{\text{cd}} = \gcd(c,d)$. The ratio of k_1 to k_2 is in its simplest form; that is, it is irreducible. So, we have

$$x = \frac{a}{b} = \frac{k_1}{k_2} \text{ and } y = \frac{c}{d} = \frac{k_1}{k_2}.$$

This allows us to write

$$\frac{a+c}{b+d} = \frac{k_1+k_1}{k_2+k_2} = \frac{k_1}{k_2} \equiv z = x = y.$$

Clearly, we can now conclude that if $x = y$ is true, then $x = z = y$ is also true.

If $x > y$, then $x > z$ and $z > y$.

This case is very similar to that for the first case (i.e., $x < y$). Basically, we can transform the reasoning for that case into the reasoning for this case by simply replacing the less-than sign in the former case by the greater-than sign in this one everywhere that it occurs. Rewriting the antecedent provides these initial equivalences:

$$\begin{aligned} x > y &\equiv ad > bc \\ &\equiv a > \frac{bc}{d} \\ &\equiv \frac{ad}{b} > c. \end{aligned} \tag{6.3.4}$$

Now, we rewrite the antecedent to provide the following additional equivalences:

$$\begin{aligned} x > y &\equiv \frac{a}{b} > \frac{c}{d} \\ &\equiv \frac{a}{b}\frac{1+\frac{d}{b}}{1+\frac{d}{b}} > \frac{c}{d}\frac{1+\frac{b}{d}}{1+\frac{b}{d}} \\ &\equiv \frac{a+\frac{ad}{b}}{b+d} > \frac{c+\frac{bc}{d}}{b+d}. \end{aligned} \tag{6.3.5}$$

Combining Equivalence 6.3.4, and Equivalence 6.3.5, and our assumption that $z = (a + c)/(b + d)$ results in

$$\frac{a+\frac{ad}{b}}{b+d} > \frac{a+c}{b+d} > \frac{c+\frac{bc}{d}}{b+d} \quad \equiv \quad x > z > y,$$

from which we can conclude that, if $x > y$ holds, then $x > z$ and $z > y$ also hold. □

The results from Lemma 6.3.1 on page 247 are an integral part of the following analyses of the three cases (i.e., $p' < q'$, $p' = q'$, $p' > q'$) for the decision-theoretic ranking method. We can use the lemma's results by mapping p', q', and t' to the lemma variables x, y, and z, respectively.

It was previously stated that the quality of ranking equation for the decision-theoretic

(DT) ranking method is

$$\mathcal{Q}'_{\mathrm{DT}} = \Pr(p' > \max(t', q')) + \Pr(p' \leq \min(t', q')). \tag{6.3.6}$$

This equation, in conjunction with the results of Lemma 6.3.1 on page 247, was used to produce Table 6.3 on page 253. For each of the 3 cases, the table lists the expressions for the minimum and maximum values of variables t' and q'; the general condition that determines the count; and the simplified counting condition after the corresponding implied condition in the first column has been taken into account. The implied conditions in the first 3 rows of the first column of the table are valid according to the cases for $x < y$, $x = y$, and $x > y$, respectively, of Lemma 6.3.1 on page 247. The above proof and the information in Table 6.3 on page 253 establish that the value of t' is always a value that is between the values of p' and q'.

The general condition for the DT ranking method is

$$(p' > \max(t', q')) \text{ or } (p' \leq \min(t', q')). \tag{6.3.7}$$

Any event in the Quadrant IV sample space, where both $p \in (0, 1)$ and $q \in (0, 1)$ hold, contributes a count of 1 if the general condition holds for that event.

The discussion in this paragraph constitutes an example of how to interpret the information in Table 6.3 on page 253. The first row of the table corresponds to the condition where $p' < q'$ holds. The implied condition (i.e., $p' < t' < q'$) for this row allows us to state that the value of $\min(t', q')$ is t' and the value of $\max(t', q')$ is q' for this row. After substituting t' for the minimum value and q' for the maximum value, we obtain the expression $(p' > q')$ or $(p' \leq t')$ in the fourth column of the first row of this table. We call this expression the *DT condition.* The last column of the first row contains the simplified version of the DT condition, that is, the expression that results after the implied condition for this row has been applied to the DT condition:

$$\begin{aligned}(p' > q') \text{ or } (p' \leq t') &= \mathit{false} \text{ or } (p' \leq t') \\ &= p' < t'.\end{aligned} \tag{6.3.8}$$

The first operand of the left-hand part of the disjunction in Equation 6.3.8 on page 251 can be replaced by the Boolean **false** value since, by De Morgan's Laws (Rosen, 1999), it is impossible for both $p' < q'$ (from the implied condition) and $p' > q'$ to be simultaneously true. Also, by repeated applications of De Morgan's laws, the condition "**false** or ($p' \leq t'$)" simplifies to the condition $p' < t'$.

Notice that the expressions in Table 6.3 on page 253, for the simplified DT conditions, look very familiar. We have certainly seen them before! More specifically, we have a counting problem which has some parts that are identical to some of the ones that we solved back in Section 5.10.3. This means that the bulk of our work has already been accomplished since we can use those results to develop the count contribution formulas for this problem.

According to the information in Table 6.3 on page 253, the contribution count for

$$p' > \max(t', q') \tag{6.3.9}$$

is the same as the contribution count for $p' > t'$ and the contribution count for

$$p' \leq \min(t', q') \tag{6.3.10}$$

is equal to the contribution count for $p' \leq t'$. Since every event in the sample space for Quadrant IV, when $p' \in (0, 1)$ and $q' \in (0, 1)$ hold, satisfies the general condition represented by Expression 6.3.7 on the preceding page, the contribution count for the expression is simply $C_4(N)$, the number of 4-compositions of N, because this expression includes every member of the set of 4-compositions of N.

If, on the other hand, we want to know the individual contribution counts for the two disjuncts of Expression 6.3.7 on page 250, that is, Expression 6.3.9 on the preceding page and Expression 6.3.10 on the previous page, then we can use the results from Section 5.10.3 to obtain the following lemmas.

Lemma 6.3.2.

$$\textit{The contribution count for } p' > \max(t', q') \textit{ is} \begin{cases} \dfrac{C_4(N) - \left|\bigcup_{1\le i\le m} G_i^{(N)}\right|}{2}, & \textit{if } N \geq 1; \\ 0, & \textit{otherwise;} \end{cases}$$

where $p' \in (0,1)$ *and* $q' \in (0,1)$.

Proof. This proof is the same as that for Lemma 5.11.6 on page 227 because, by the information in Table 6.3 on the next page, if $p' > \max(t', q')$ is true, then $p' > t'$ must be also true. □

Lemma 6.3.3.

$$\textit{The contribution count for } p' \leq \min(t', q') \textit{ is} \begin{cases} C_4(N) - \mathrm{cc}_{p'>\max(t',q')}, & \textit{if } N \geq 1; \\ 0, & \textit{otherwise;} \end{cases}$$

where $p \in (0,1)$, $q \in (0,1)$, *and* $\mathrm{cc}_{p'>\max(t',q')} = \frac{C_4(N) - \left|\bigcup_{1\le i\le m} G_i^{(N)}\right|}{2}$.

Proof. As was stated earlier, all of the members of the set of 4-compositions for N satisfy 6.3.7 on page 250. Since exactly one of the conditions $p' < t'$, $p' = t'$, $p' > t'$ holds for an arbitrary member of this set, the contribution count for $p' \leq \min(t', q')$ is the difference between $C_4(N)$ and $\mathrm{cc}_{p'>\max(t',q')}$, the contribution count for $p' > \max(t', q')$. □

Table 6.3: The Three Cases for the Decision-Theoretic (DT) Condition in Quadrant IV.

case condition	$\min(t', q')$	$\max(t', q')$	DT condition	DT condition (simplified)
$p' < q'$ (implies $p' < t' < q'$)	t'	q'	$(p' > q')$ or $(p' \leq t')$	$p' < t'$
$p' = q'$ (implies $p' = t' = q'$)	t' (or q')	t' (or q')	$(p' > t')$ or $(p' \leq t')$	$p' = t'$
$p' > q'$ (implies $p' > t' > q'$)	q'	t'	$(p' > t')$ or $(p' \leq q')$	$p' > t'$

From the information in Table 4.11 on page 140 – with the exception of the case in Quadrant IV where the conditions $p \in (0, 1)$ and $q \in (0, 1)$ are simultaneously true – we can calculate the count contributions for $(p' > \max(t', q'))$ and $(p' \leq \min(t', q'))$ where $N \geq 1$. The expression for the former condition is

$$4N - 1 + 2\binom{N-1}{2} + \frac{C_4(N) - \left|\bigcup_{1 \leq i \leq m} Q4C_i^{(N)}\right|}{2} \tag{6.3.11}$$

and the one for the latter one is

$$N - 1 + 2\binom{N-1}{2} + C_4(N) - \frac{C_4(N) - \left|\bigcup_{1 \leq i \leq m} Q4C_i^{(N)}\right|}{2}. \tag{6.3.12}$$

After combining Expressions 6.3.11 and 6.3.12, we obtain a total contribution of

$$\begin{aligned}
& 5N - 2 + 4\binom{N-1}{2} + C_4(N) \\
= \; & 5N - 2 + 4\binom{N-1}{2} + \binom{N-1}{3} \\
= \; & \frac{6(5N-2)}{6} + 4 \cdot \frac{3}{3} \cdot \frac{(N-1)(N-2)}{2!} + \frac{(N-1)(N-2)(N-3)}{3!} \\
= \; & \frac{30N - 12}{6} + \frac{12(N-1)(N-2)}{6} + \frac{(N-1)(N-2)(N-3)}{6} \\
= \; & \frac{30N - 12 + 12(N-1)(N-2) + (N-1)(N-2)(N-3)}{6} \\
= \; & \frac{30N - 12 + 12(N^2 - 3N + 2) + N^3 - 3N^2 + 2N - 3N^2 + 9N - 6}{6}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{30N - 12 + 12N^2 - 36N + 24 + N^3 - 3N^2 + 2N - 3N^2 + 9N - 6}{6} \\
&= \frac{N^3 + 6N^2 + 5N + 6}{6} \\
&= \frac{(N^3 + 6N^2 + 5N + 6) + 6N}{6} - \frac{6N}{6} \\
&= \frac{N^3 + 6N^2 + 11N + 6}{6} - N \\
&= \frac{(N+3)(N+2)(N+1)}{6} - N \\
&= \frac{(N+3)(N+2)(N+1)}{3!} - N \\
&= \binom{N+3}{3} - N.
\end{aligned}
$$

With this result, we can state that

$$
\begin{aligned}
\mathcal{Q}'_{\mathrm{DT}} &= \Pr(p' > \max(t', q')) + \Pr(p' \leq \min(t', q')) \\
&= \frac{\binom{N+3}{3} - N}{\binom{N+3}{3}}. \qquad (6.3.13)
\end{aligned}
$$

6.3.1 Mean and Variance

Definition 6.3.1.1. Given a sample space Ω, the discrete random variable $\mathcal{B}$ is a function such that, for each outcome $\omega \in \Omega$,

$$
\mathcal{B}(\omega) = \mathrm{BooleToNat}((p'_\omega > \max(t'_\omega, q'_\omega)) \text{ or } (p'_\omega \leq \min(t'_\omega, q'_\omega))),
$$

where Ω is the set of weak 4-compositions of N, and p'_ω and q'_ω are the p' and q' values, respectively, for this ω.

The last two definitions can be used to rewrite $\mathcal{Q}'_{\mathrm{DT}}$ to show that $\mathcal{Q}'_{\mathrm{DT}}$ is simply the expected value, or mean, of $\mathcal{B}$.

Lemma 6.3.4. *The expected value of* $\mathcal{B}$ *is* $\mathcal{Q}'_{\mathrm{DT}}$. *That is,*

$$\begin{aligned} E[\mathcal{B}] &= \sum_{\omega\in\Omega} \mathcal{B}(\omega)\binom{N+4-1}{4-1}^{-1} \\ &= \frac{\sum_{\omega\in\Omega}\mathcal{B}(\omega)}{\binom{N+3}{3}} \\ &= \mathcal{Q}'_{\mathrm{DT}}. \end{aligned}$$

Proof. The number of weak 4-compositions for N is $\binom{N+4-1}{4-1}$, by Equation 2.2.2 on page 26. These compositions are equally likely, each with a probability of $\binom{N+4-1}{4-1}^{-1}$. The random variable $\mathcal{B}$ is binary-valued, with values that are either 0 or 1; i.e., $\mathcal{B}(\omega) \in \{0, 1\}$. Each weak 4-composition of N, denoted by ω, where the p' value is greater than the corresponding t' value, is associated with a $\mathcal{B}(\omega)$ value of 1; otherwise, the $\mathcal{B}(\omega)$ value is 0. □

Lemma 6.3.5. *The variance of* $\mathcal{B}$ *is*

$$Var[\mathcal{B}] = E[\mathcal{B}^2] - (E[\mathcal{B}])^2 \tag{6.3.14}$$

$$= E[\mathcal{B}] - (E[\mathcal{B}])^2 \tag{6.3.15}$$

$$= \mathcal{Q}'_{\mathrm{DT}} - (\mathcal{Q}'_{\mathrm{DT}})^2 \tag{6.3.16}$$

$$= \mathcal{Q}'_{\mathrm{DT}}(1 - \mathcal{Q}'_{\mathrm{DT}}). \tag{6.3.17}$$

Proof. Line 6.3.14 follows directly from the concept of variance in Definition 5.12.0.1 on page 228. The value of $\mathcal{B}(\omega)$ is always the same as that of $(\mathcal{B}(\omega))^2$ because, by Definition 6.3.1.1 on the previous page, $\mathcal{B}(\omega)$ can only take on the values 0 and 1. This is the justification for Line 6.3.15. The justification for Line 6.3.16 comes from Lemma 6.3.4 because it established that $\mathcal{Q}'_{\mathrm{DT}}$ is the expected value of $\mathcal{B}$, that is, $\mathcal{Q}'_{\mathrm{DT}} = E[\mathcal{B}]$. The expression in Line 6.3.17 is a basic factoring of the expression in Line 6.3.16. □

Table 6.4: Number of Qualifying Contributions ($1 \leq N \leq 20$).

	number of	total number of	$\mathcal{Q}'$	$\sqrt{\mathcal{Q}'(1-\mathcal{Q}')}$
N	qualifying weak 4-comps	weak 4-comps	(mean)	(standard deviation)
1	3	4	0.75	0.433 013
2	8	10	0.8	0.4
3	17	20	0.85	0.357 071
4	31	35	0.885 714	0.318 158
5	51	56	0.910 714	0.285 156
6	78	84	0.928 571	0.257 539
7	113	120	0.941 667	0.234 373
8	157	165	0.951 515	0.214 788
9	211	220	0.959 091	0.198 08
10	276	286	0.965 035	0.183 691
11	353	364	0.969 78	0.171 192
12	443	455	0.973 626	0.160 244
13	547	560	0.976 786	0.150 583
14	666	680	0.979 412	0.142 001
15	801	816	0.981 618	0.134 33
16	953	969	0.983 488	0.127 433
17	1123	1140	0.985 088	0.121 202
18	1312	1330	0.986 466	0.115 545
19	1521	1540	0.987 662	0.110 388
20	1751	1771	0.988 707	0.105 667

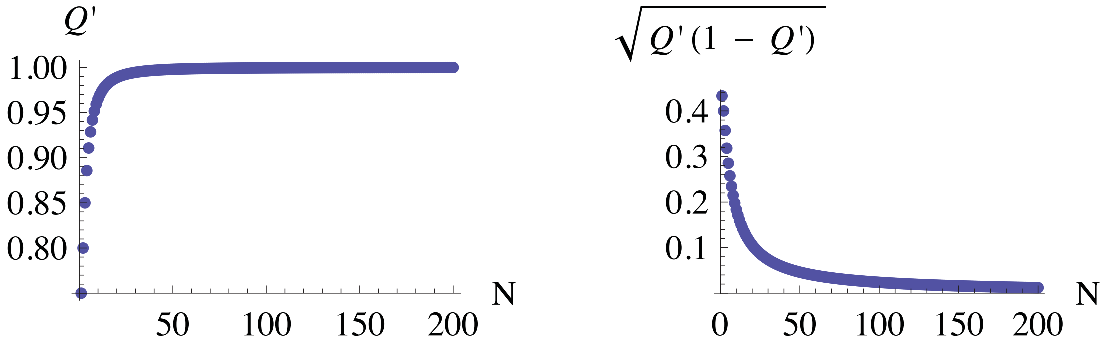


Figure 6.2: Plot of the mean ($\mathcal{Q}'$) and standard deviation $\left(\sqrt{\mathcal{Q}'\,(1-\mathcal{Q}')}\right)$ for $\mathcal{B}$ for the DT ranking method when $1 \leq N \leq 200$.

6.4 Summary for the DT Ranking Method

The last section in this chapter contains a combinatoric model of $\mathcal{Q}'$ for the DT ranking method. Similar to the development of the counting expressions for the coordination level matching and inverse document frequency ranking methods, this section used the results of the analyses and associated formulas from Chapter 4 to develop some of the counting expressions that were used in these sections. Except for the situation where the joint conditions $p' \in (0, 1)$ and $q' \in (0, 1)$ were both true, no derivation work had to be done in this chapter. For the situation just mentioned, these sections used mathematical and combinatorial arguments and techniques to develop the counting expressions that applied to it.

The relevant expressions from Chapter 4 were combined with those that were developed in this chapter. After simplifying these expressions, the result was a closed-form expression.

Figure 6.2 contains mean and the standard deviation plots for the DT ranking method. Table 6.4 on the previous page contains the values that the first 20 points of these plots are based on. Research for this chapter indicated that the ordinate (i.e., y-axis) asymptote was 1 for the first plot and was 0 for the second plot. The research also showed that the plotted values were monotonically increasing in the first plot as the number of documents in a collection increases from 1 to ∞ (infinity) and were monotonically decreasing in the second plot as the number of documents increased.

Chapter 7

Characteristics of a Combinatoric-Based $\mathcal{A}$ and ASL Performance Measure

This chapter addresses the first of the three research questions that were enumerated in Section 3.5, which starts on page 103: What would be the characteristics of a combinatoric measure (CM_ASL), based on the Average Search Length (ASL), that performs the same as a probabilistic measure of retrieval performance, also based on the ASL? More specifically, Section 3.5.1 contains the initial introduction for this research question. The ASL measures that are developed in this chapter are used to help with the validation efforts that are discussed in Chapter 8 and with the answering of the second research question in Chapter 9.

A central item of interest with respect to each document in a collection is whether a particular feature is present or absent in that document. If we assume that the documents are textual (as contrasted to other multimedia types such as image, video, audio, graphics, or animation) then "features can be keywords, phrases, or structural elements" (Rui et al., 1999). The number of times that a feature (e.g., term, word, phrase) occurs in a document is called its *feature frequency*. If, for example, the term "shoe" occurs five times in a particular document, the feature frequency for "shoe" equals 5 for that document.

In this chapter, the concept *term* is always synonymous with the intuitive concept of *word*. "North Carolina", for example, is not considered a term. Instead, it is viewed as a phrase that consists of two terms (i.e., "North" and "Carolina"). Also, in this chapter, both relevance and feature frequency are represented as binary values. For relevance, this means that a term is either relevant or not relevant. For feature frequency, if a feature (e.g., term, word) is present (i.e., occurs one or more times) in a document, it is said to have feature frequency 1 regardless of how many times it actually occurs; otherwise, it is absent and is said to have feature frequency 0.

It is very important to note that the models that are developed in this chapter can handle binary relevance but cannot handle continuous relevance. The remainder of this chapter details how binary relevance is incorporated into these models.

Many mathematical concepts were introduced in the previous chapters and used to help derive many of the equations that first appeared in these chapters. This chapter also introduces several additional concepts that are crucial to the derivations that take place in his chapter. Among these prior concepts, and those from this chapter, are compositions (Andrews, 1984; Andrews and Eriksson, 2004), partitions (Andrews, 1984; Andrews and Eriksson, 2004), the Principle of Inclusion-Exclusion (Andrews, 1984), the greatest common divisor (Rosen, 2005), the power set of a set (Rosen, 1999; Rosen et al., 2000), permutations (Riordan, 1958), and combinations (Riordan, 1958). This chapter also introduces the statistical concepts of expected value (Terrell, 1999) and variance (Terrell, 1999). The main mathematical concepts that are introduced in this chapter are Gaussian polynomials (Andrews, 1984), probability mass functions (Graham et al., 1994), generating functions (Riordan, 1958; Graham et al., 1994; Charalambides, 2002), and probability generating functions (Riordan, 1958).

Of course, all of these concepts have a much wider sphere and range of applicability than the uses that they were put to in the previous chapter and are put to in this chapter and the subsequent ones. It might be helpful to point out some of these additional uses. This is the purpose of the next few paragraphs.

One large application area for partitions is in statistics, particularly non-parametric statistics (Barton, 1959; David, 1959; Harary, 1959). There, the interest is often "in restricted partitions, that is, partitions in which the largest part is, say $\leq N$ and the number of parts is $\leq M$. This ... will naturally lead ... to the Gaussian polynomials and from there to ... permutations" (Andrews, 1984). Compositions, combinations, the Principle of Inclusion-Exclusion, probability mass functions, generating functions, and probability generating functions are also used very heavily in statistical theory (Barton, 1959; David, 1959; Harary, 1959; Johnson et al., 2005; Charalambides, 2005) and in the analysis of algorithms (Dobrushkin, 2009). Generating functions are the "most important idea in enumerative combinatorics" (Gessel, 1985) and are also heavily used in mathematical statistics (Terrell, 1999; Rose and Smith, 2002) and applied combinatorics (Tucker, 1980; Gross, 2008; Roberts and Tesman, 2009).

The Gaussian polynomials, also known as the Gaussian binomials (or the Gaussian coefficients or the q-binomial coefficients), are a generalization of the binomials (Gasper and Rahman, 2004). Mathematicians refer to these q-binomial coefficients as the q-analogs of the binomial coefficients and they are part of an important class of series known as the q-series (or q-hypergeometric series or basic hypergeometric series) (Gasper and Rahman, 2004). The q-series has wide applicability in many mathematical areas, including analysis, number theory, combinatorics, physics, and computer algebra (Andrews, 1974, 1986; Fine, 1988; Berndt and Ono, 2001; Rakha and El-Sedy, 2004; Charalambides, 2005; Johnson et al., 2005).

7.1 Notation and Definitions

Notationally, let $N = R+S$, where $R = r_0+r_1$ and $S = s_0+s_1$, represent the total number of documents in a non-empty (i.e., $N > 0$) collection with $R = r_0 + r_1$ representing the number of *relevant* documents and $S = s_0 + s_1$ representing the number of non-*relevant* documents. In this collection, there are n_0 total documents with feature frequency 0, n_1 total documents with feature frequency 1, r_0 relevant documents with feature frequency 0, r_1 relevant documents with feature frequency 1, s_0 non-relevant documents with feature frequency 0, and s_1 non-relevant documents with feature frequency 1. An ordering is represented by a sequence of N documents. By definition, in an optimal ordering, all the n_1 documents with feature frequency 1 appear before any of those with feature frequency 0 (Losee, 1998). Of the N possible positions in the ordering that a document can appear in, it is assumed that a document is equally likely to occupy any of these positions but does not share that same position with any other document in the same ordering. Put another way, each of the N documents is associated with exactly one position in a specific ordering of N documents and each of the N positions is associated with exactly 1 document. Mathematically, the mapping between documents and positions is a bijection (Rosen, 1999). Some of the definitions from Chapter 2 are repeated below because they are used later in this chapter.

The variable $\mathcal{A}$ (normalized search length) is computed by noting that documents with feature frequency 1 are at the low end of the $\mathcal{A}$ spectrum (good performance) and those with feature frequency 0 are at the high end of the spectrum (poor performance). Let d denote the random variable whose value is 1 for a document if the document contains the query term (i.e., its feature frequency is 1) and 0 if the document does not contain the query term (i.e., its feature frequency is 0). Therefore, $\Pr(d = 1)$ denotes the probability that a document in a collection for a query q contains the query term and $\Pr(d = 0)$ denotes the probability that a document does not contain the query term. Of course,

$$\Pr(d = 1) = 1 - \Pr(d = 0).$$

Since d is binary-valued, let d denote the same meaning as $d = 1$ and let $\overline{d}$ denote the same meaning as $d = 0$. This helps to simplify the notation that is used in subsequent discussions.

We can use these denotations to state that the formula for the normalized search length is

$$\mathcal{A} = \frac{1 + \Pr(d) - \Pr(d|rel)}{2}.$$

Notationally, the equation can be simplified by letting $p = \Pr(d|rel)$ (the probability that a relevant document has a feature frequency of 1) and $t = \Pr(d)$ (the probability that any document has a feature frequency of 1):

$$\mathcal{A} = \frac{1 + t - p}{2}. \tag{7.1.1}$$

The formula for the Average Search Length (ASL) is

$$\text{ASL} = N\left(\mathcal{Q}\mathcal{A} + \overline{\mathcal{Q}}\;\overline{\mathcal{A}}\right) + 1/2. \qquad \text{(Losee, 1998)}$$

Briefly, in the above equation for the Average Search length, N is the number of documents to be ranked, $\mathcal{Q}$ is the probability that the ranking is optimal, and $\mathcal{A}$ is the normalized expected position of a relevant document from the front of the ranking. In the above formula, $\overline{\mathcal{A}}$ is defined as $1 - \mathcal{A}$ and $\overline{\mathcal{Q}}$ is defined as $1 - \mathcal{Q}$. Both $\mathcal{Q}$ and $\mathcal{A}$ are values in the closed interval $[0, 1]$.

7.2 A Combinatoric Model of $\mathcal{A}$

$\mathcal{A}$ is defined as the normalized average position of a relevant document from the front of an ordered list (Losee, 1998). The normalized positions are in the closed interval $[0, 1]$. A document at the front of the list would have a normalized position of 0; a document at the back of the list would have a normalized position of 1.

The computation of $\mathcal{A}$ can be for a single ordering or it can be extended to calculate the value for a set of orderings (such as those associated with all the possible orderings for a collection of N documents). This section of the chapter is concerned with the calculation of $\mathcal{A}$ for the latter situation.

We proceed to calculate $\mathcal{A}$ by essentially a two-step process: compute the unnormalized average value for $\mathcal{A}$ over all of the possible N-document orderings when s_0, s_1, r_0, and r_1 are known; then normalize this value so that it is in the closed interval $[0, 1]$.

Figure 7.1 on the next page describes an optimal ordering of N documents in terms of the parameters N, s_0, s_1, r_0, and r_1. Note that the positions associated with those documents are unnormalized (for the time being). Figure 7.2 on page 266 imparts concreteness to the abstractness associated with Figure 7.1 on the next page. It is an example that lists the sample space for an 8 document collection for specified values of r_1, r_0, s_1, and s_0. Each row represents a sample point (i.e., a possible sequence of documents in an optimal ordering) and the column numbers represent document positions within a sequence. Taken together, the rows constitute an exhaustive enumeration of all of the sample points in the sample space.

7.2.1 An example of a sample space for an optimal ordering of 8 documents

Consider the sample space depicted in Figure 7.2 on page 266. The parameters of the collection that it represents are $N = n_1 + n_0 = 8$, $n_1 = r_1 + s_1 = 3$, $n_0 = r_0 + s_0 = 5$,

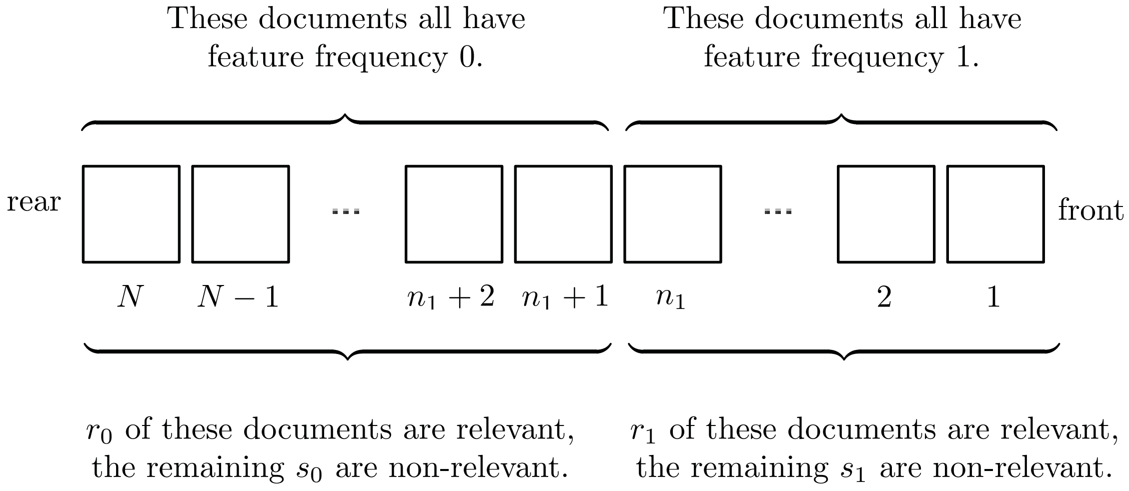

These documents all have feature frequency 0.
These documents all have feature frequency 1.
rear
front
N
$N-1$
n_1+2
n_1+1
n_1
2
1
r_0 of these documents are relevant, the remaining s_0 are non-relevant.
r_1 of these documents are relevant, the remaining s_1 are non-relevant.

Figure 7.1: This depicts an optimal ordering of N documents. Each of the squares represents a position in the ordering. The front of the ordering is defined to be at position 1; the rear of the ordering is at position N.

with $r_1 = 2$, $s_1 = 1$, $r_0 = 1$, and $s_0 = 4$. Documents with feature frequency 1 occupy positions 1-3, inclusive, whereas documents with feature frequency 0 occupy positions 4-8, inclusive. From Figure 7.2 on the following page, one can readily see that there are three unique sample points for the documents with feature frequency 1 and five unique sample points for those with feature frequency 0. Jointly, there are $3 \times 5 = 15$ sample points in the sample space when both feature frequencies are involved.

In IR terms, the sample space illustrates the 15 different rankings that are possible for an 8 document collection that has 2 distinct retrieval status values (RSVs). The higher-valued RSV has three documents that are associated with it, of which two are of one kind (i.e., relevant) and the other is of another kind (i.e., non-relevant). The lower-valued RSV has 5 documents that are associated with it, with one document being of one kind (i.e., relevant) and the remaining four being of another kind (i.e., non-relevant).

The information in the figure, and in this example, is related to whether a sequence of ranked documents is weakly ordered (i.e., some of the RSVs are duplicates of other RSVs

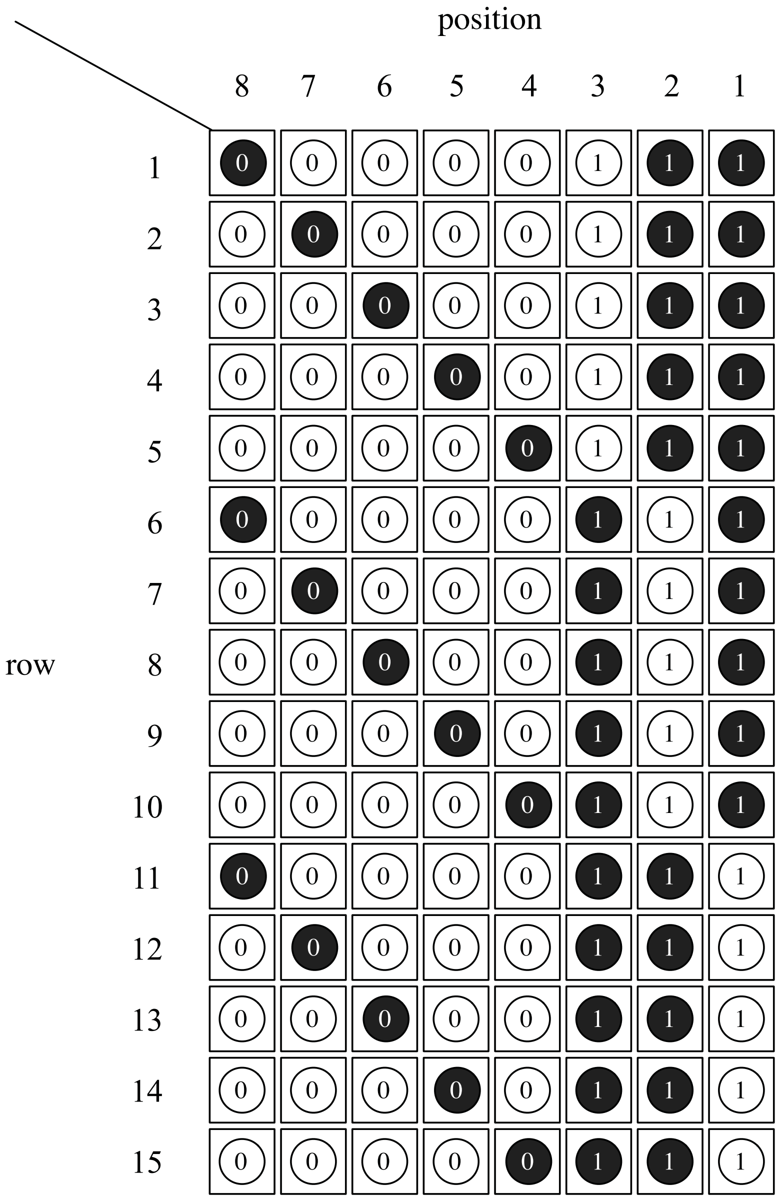


Figure 7.2: This diagram details each of the 15 possible sample points that can occur in the sample space that is associated with an optimal ordering of 8 documents (i.e., $N = 8$), with 5 of the documents having feature frequency 0 and 3 of them having feature frequency 1. Of those documents with feature frequency 0, one is relevant (i.e., $r_0 = 1$) and four are non-relevant (i.e., $s_0 = 4$) . The documents with feature frequency 1 have two that are relevant (i.e., $r_1 = 2$) and one that is non-relevant (i.e., $s_1 = 1$). In this diagram, each of the dark background balls represents a single relevant document whereas each of the light background balls represents a single non-relevant document. The number inside each ball represents the feature frequency of the document associated with that ball. Each row represents an ordering of these balls, each column represents a position in the ordering. Position 1 is defined to be the front of an ordering, position 8 is the rear of an ordering.

which means that ties are present) or are strongly ordered (i.e., the RSVs are distinct or unique). The notions of strong and weak orders are very important during performance evaluation. Failure for a performance measure to take these kinds of orders into account can result in erroneous, or misleading, results from a performance evaluation. For example, if a performance evaluation, that uses the ASL, assumes that the rankings from the document collection with the parameters that were specified in the first paragraph of this subsection is strongly ordered, when in actuality it is weakly ordered, then the ASL that it calculates can range from a *minimum* value of 7/3 (i.e., the line 5 ranking in Figure 7.2 on the previous page has three relevant documents at positions 1, 2, and 4 for a sum of positions that is equal to 7) to a *maximum* value of 13/3 (i.e., the line 11 ranking has three relevant documents at positions 2, 3, and 8 for a sum of positions that is equal to 13). These values can be contrasted with the value of 17/6 that the ASL would calculate if it took into account that the documents are weakly ordered. Section 10.2.2 and Section 10.3 contain a detailed discussion of weak orders, strong orders, and how to develop performance measures whose calculated values are consistent with the assumption that *some* of the documents in a vector V of ranked documents *may* have tied (i.e., duplicate) RSVs. Measures of this type are referred to by the adjective *Type-T* in Chapter 10 and, in that chapter, we develop Type-T versions of the ASL, ESL, MZE, RR, recall, and precision measures.

7.2.2 Permutations, permutation trees, r-permutations, and r-combinations

A very important aspect of the calculation of $\mathcal{A}$ is being able to analytically determine the unnormalized average position of a relevant document in an arbitrary sample space. Given a method to determine the distribution of the positions of relevant documents in a sample space, we can very easily calculate the average position. Central to these calculations is knowledge of the mathematical concepts of permutations, combinations, and permutation trees. Brief descriptions of each of these follow in the next few paragraphs. Those descriptions provide much of the foundation that we use to later prove two theorems about the distribution of objects in a r-permutation.

A *permutation* of a set of n distinct objects (i.e., document positions) is an ordered arrangement of these objects. For example, if there are three distinct objects named A, B, and C, then there are $3! = 6$ ways of ordering them: ABC (A first, B second, and C third), ACB, BAC, BCA, CAB, CBA. Sometimes, we are interested in ordering $r \leq n$ of them where r may be less than n. An ordered arrangement of r members is called an r-permutation. The 2-permutations of A, B, and C are AB, AC, BA, BC, CA, and CB. The 1-permutations are A, B, and C. The number of r-permutations of a set of n distinct objects is $P(n, r) = n(n-1)\ldots(n-r+1)$ where $r \leq n \leq 1$. In the discussions to follow, it is assumed that $n, r \in \mathbb{Z}^+$ where $\mathbb{Z}^+$ denotes the set of positive integers.

An r-combination of a set of n distinct objects is an unordered selection of r members from the set. In other words, an r-combination is simply a subset of the set with r objects. The number of r-combinations of a set of n distinct objects is $C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$ where $r \leq n \leq 1$. The 3-combination of A, B, and C is ABC (because it and ACB, BAC, BCA, CAB, CBA are all equivalent because order does not matter); the 2-combinations are AB, AC, BC; and the 1-combinations are A, B, and C.

A *permutation tree* (Takaoka, 1999; Arce and Tian, 1996; Trippi, 1975) models the choices available when forming a permutation (Figure 7.3 on the following page shows a permutation tree for 4 objects). Permutations can be visualized as paths in this tree. In the literature, there are two main ways to represent a permutation tree: (1) the labels on the edges represent the choices or (2) the labels on the non-root nodes represent the choices. Depending on the situation, each representation has its respective advantages and disadvantages. We use representation (2) in this document. The root node in this

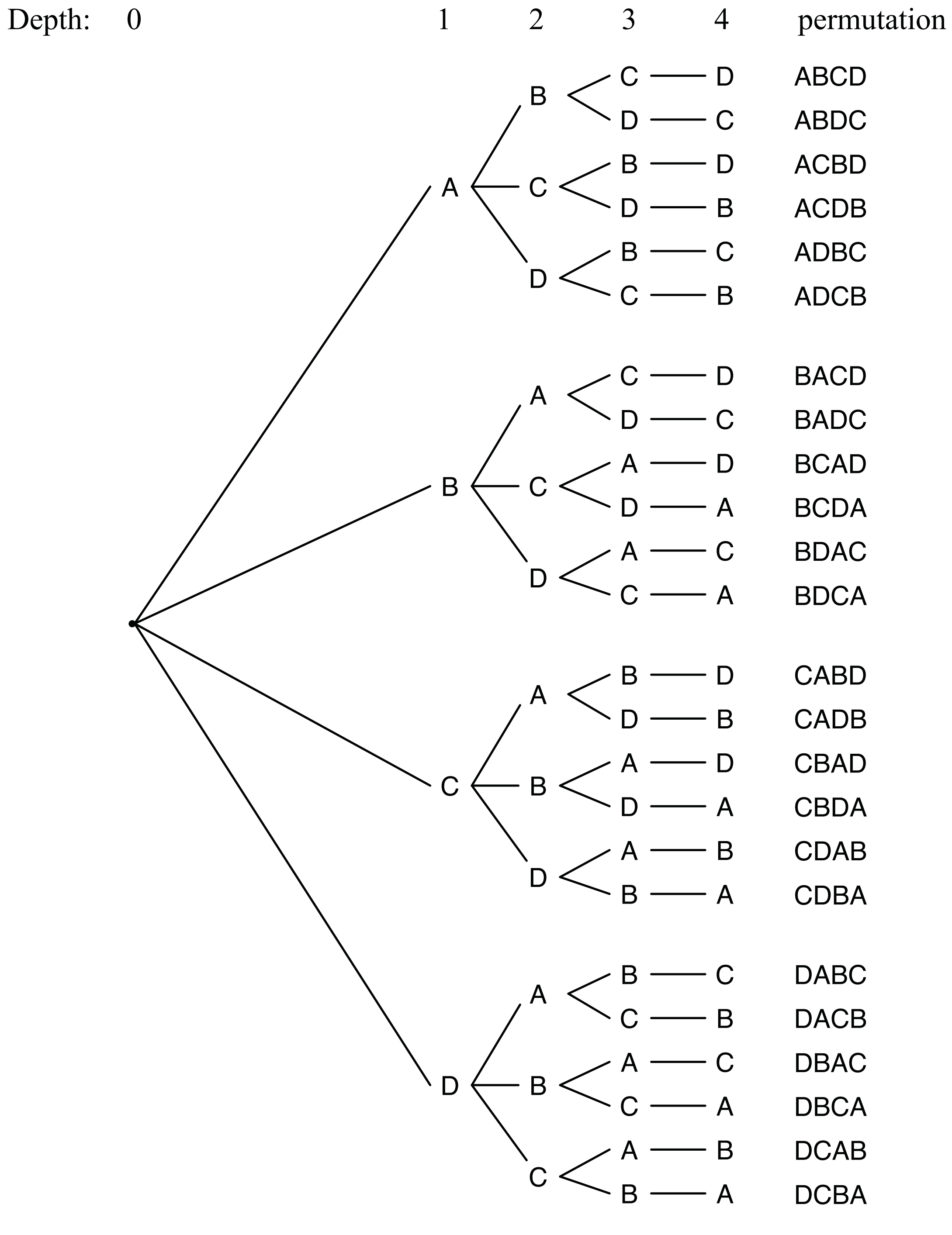


Figure 7.3: A permutation tree for 4 distinct objects named A, B, C, and D.

representation does not represent any choice, it is just a starting point for the generation process that is described below.

The labels on the non-root nodes in a permutation tree are surrogates for the objects that are being permuted. Each unique object has a label that is different from that of any other object. The labels on the nodes at level 1 of the permutation tree denote the individual objects that are being permuted. A node label corresponds to an object contained in the set $O = \{o_1, o_2, \ldots, o_n\}$ with $|O| = n$. To simplify matters, unless stated otherwise, labels and the objects they denote have the same name. That is, if an object has the name o_1, its label is also o_1. Choosing a node in the permutation tree is equivalent to selecting an object, and permutations are formed by following paths from the root to the leaves. The concatenation of the n non-root node labels along the path from the root to a leaf in the tree is the permutation that corresponds to that path.

A permutation tree of n objects is generated recursively; at the root all n objects are available for selection, so the root has n edges (one for each object) leading to n different subtrees. At the subtrees of the root only $n-1$ objects are available (because one object has already been selected), so the subtrees of the root each have $n-1$ edges leading to $n-1$ subtrees. The missing depth 2 object in each subtree corresponds to the the object that has been selected at depth 1, and each subtree itself is a permutation tree for $n-1$ objects. This process is repeated with the subtrees of the subtrees, and so on, until the leaves are reached. At that point all objects have been selected; therefore, no objects are available to continue the process.

This manner in which the the permutation tree is constructed guarantees that all permutations of n objects correspond to paths in the tree. Also, no two distinct paths correspond to the same permutation of objects. This implies that there are as many permutations of n objects as there are paths in a permutation tree for n objects.

Determining the number of paths in a permutation tree is relatively easy. Since n

paths emerge from the root (i.e., the number of ways that one of n objects can be chosen), depth 1 has n nodes. Each of these nodes is the root of a subtree that splits into $n-1$ nodes, each of these $n-1$ nodes is the root of a subtree that splits into $n-2$ nodes and so on. The original n paths terminate at the leaves by which time they have split into $n(n-1)\dots 2\cdot 1 = n!$ paths (exactly one for each possible permutation).

Now suppose that instead of generating a permutation tree for n distinct objects, we want to generate one for $1 \leq r \leq n$ objects. This corresponds to the generation of r-permutations.

The use of a tree to represent the set of r-permutations for n distinct objects proceeds similarly to that to generating full permutations (i.e., when $r = n$) and can be accomplished in the following way: Initially, create a tree with a single node. This is the root of the tree. Next, create n child nodes of the root. Each of these depth 1 nodes corresponds to a different object. One can view these depth 1 nodes as representing the number of ways that one object can be chosen out of a collection of n distinct objects. The number of ways of doing this is the same as the number of paths from the root to the newly created depth 1 nodes. This value is $P(n,1)$, or after simplifying, just n.

Assuming that $n \geq 2$, we can extend the tree in the following way to show all the number of ways that one can generate permutations of two distinct objects. Each node at depth 1 has $n-1$ depth 2 children. These child nodes represent all the objects in O, except for the one that is represented by the parent node (i.e., the depth 1 node that is the parent of these children). The number of ways of doing this is the same as the number of paths from the root to the newly created depth 2 nodes. This value is $P(n,2)$ because each of the n depth 1 nodes in the tree has $n-1$ children.

In general, a tree to represent r-permutations for a specific value of r has a depth of r (shown by Figure 7.4 on the next page). Each depth d node, where $0 \leq d < r$, has $n-d$ depth *d+1* nodes as children. Let $L = \{l_1, l_2, \dots, l_n\}$ be the labels for the objects in O and let there be a bijection f between L and O. Additionally, let L_x be the set of object labels on the path from the root to an arbitrary node x at depth d in the tree and let $L'_x = L - L_x$ be the set of $n - d$ object labels that do not appear along that path. To extend the part of the tree, that has node x as its parent, to depth $d + 1$, create $n - d$ children for node x. Each of these children represents a different one of the object labels in L'. The total number of depth $d + 1$ nodes in the tree is $P(n, d + 1)$.

All of the possible choices at each depth build on the choices at the immediately preceding depth. The number of paths to depth r nodes is $n(n - 1) \dots (n - r + 2)(n - r + 1) = P(n, r)$. If we substitute $r - 1$ for r in the previous equation, we obtain $P(n, r - 1) = n(n - 1) \dots (n - (r - 1) + 1) = n(n - 1) \dots (n - r + 2)$. This implies that $P(n, r) = (n - r + 1)P(n, r - 1)$ which means that each node at depth $2 \le d < r$ has $n - l + 1$ child nodes at depth $d + 1$.

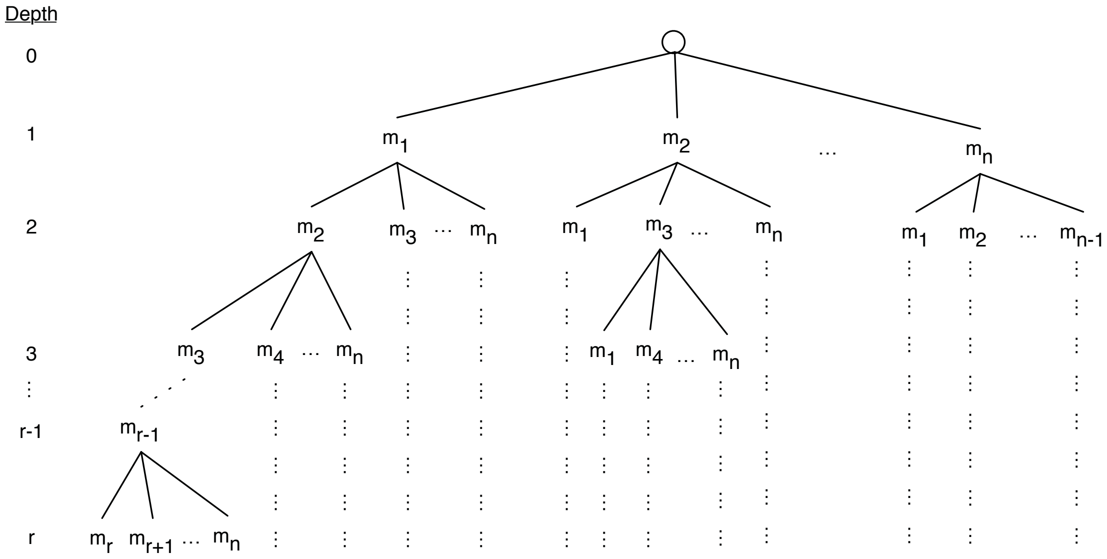


Figure 7.4: A generalized version of a permutation tree for r-permutations.

Theorem 7.2.1. *Each of the $n \ge r \ge 1$ distinct members in set $O = \{o_1, o_2, \dots, o_n\}$ occurs $\frac{rP(n,r)}{n}$ times in the sample space of $P(n, r)$ sample points of r-permutations for* *that set.*

Proof. The proof of this theorem uses induction and arguments that are based on the manner in which an r-permutation tree is constructed. Earlier discussions established that any sample space for r-permutations can be represented by such a tree. Let $\text{Freq}(n, r) = \frac{rP(n,r)}{n}$ and let $\text{Prop}(n, r, O)$ be the proposition (i.e., claim or assertion) that each of the members in set O occurs $\text{Freq}(n, r)$ times in the sample space for O.

Case $n = r = 1$.

This means that $\text{Prop}(1, 1, O)$ is true because the root has only one descendant and $\frac{rP(n,r)}{n} = \frac{1*P(1,1)}{1} = \frac{1*1}{1} = 1$.

Case $n \geq 2$.

Basis step. $\text{Prop}(n, 1, O)$ is true because $r = 1$ means that there is only one level in the tree below the root and that these n distinct child nodes labeled $o_1, o_2, \ldots, o_n$ have the root as their parent. Hence, each node labeled o_i occurs $\frac{rP(n,r)}{n} = \frac{1*P(n,1)}{n} = \frac{n}{n} = 1$ times because level 1 of the tree only has one instance of each of them.

Inductive step. Assume that $\text{Prop}(n, l, O)$ is true for $2 \leq l < r$. We can extend the permutation tree from one that has $r - 1$ levels to one that has r levels in the following way such that $\text{Prop}(n, r, O)$ is also true.

Without loss of generality, let $p = o_1 o_2 \ldots o_{r-1}$ be an arbitrary r–1-permutation of the n members of set O. Let set $C_p = \{o_1, o_2, \ldots, o_{r-1}\}$ be the collection of $r - 1$ objects that appear in p. Compute $C'_p = O - C_p$. The set C'_p contains $n - (r - 1) = n - r + 1$ members, each of them corresponding to one of the objects in O that does not appear in C_p.

All of the possible r-permutations of set O, that have p as a prefix, can be generated from this r–1-permutation by the three steps that are enumerated below. An

r-permutation that has prefix p is a permutation where the length of its sequence of objects (i.e., the number of objects in the permutation) is one more than $r-1$ (the length of the prefix p). In essence, the first $r-1$ objects in such an r-permutation are the same as those that are in the permutation $p = o_1 o_2 \ldots o_{r-1}$. Furthermore, any given object o has the same position in the r-permutation as it does in the $r-1$-permutation p.

1. Make $|C'_p| = n - r + 1$ copies of the permutation p.
2. Visit each of these $n - r + 1$ instances once.
3. Perform the following actions at each visit:
   (a) Remove one member from C'_p, thereby leaving one less member in this set than it had prior to visiting the current instance.
   (b) Append it to the right end of the current instance.

The result of applying this process to an arbitrary r–1-permutation p of set O is the set of all r-permutations that have p as a prefix (shown by Figure 7.5(a)) on the following page. The cardinality of this set is $n - r + 1$. Collectively, the result of applying this process to the set of all r–1-permutations of O is the set of all r-permutations of O. The cardinality of this set is

$$\begin{aligned}(n-r+1)P(n,r-1) &= (n-r+1)n(n-1)(n-2)\ldots(n-(r-1)+1)\\ &= n(n-1)(n-2)\ldots(n-(r-1)+1)(n-r+1)\\ &= n(n-1)(n-2)\ldots(n-r+2)(n-r+1)\\ &= P(n,r)\end{aligned}$$

and the r-permutations in its sample space have a total of $rP(n,r)$ object occurrences.

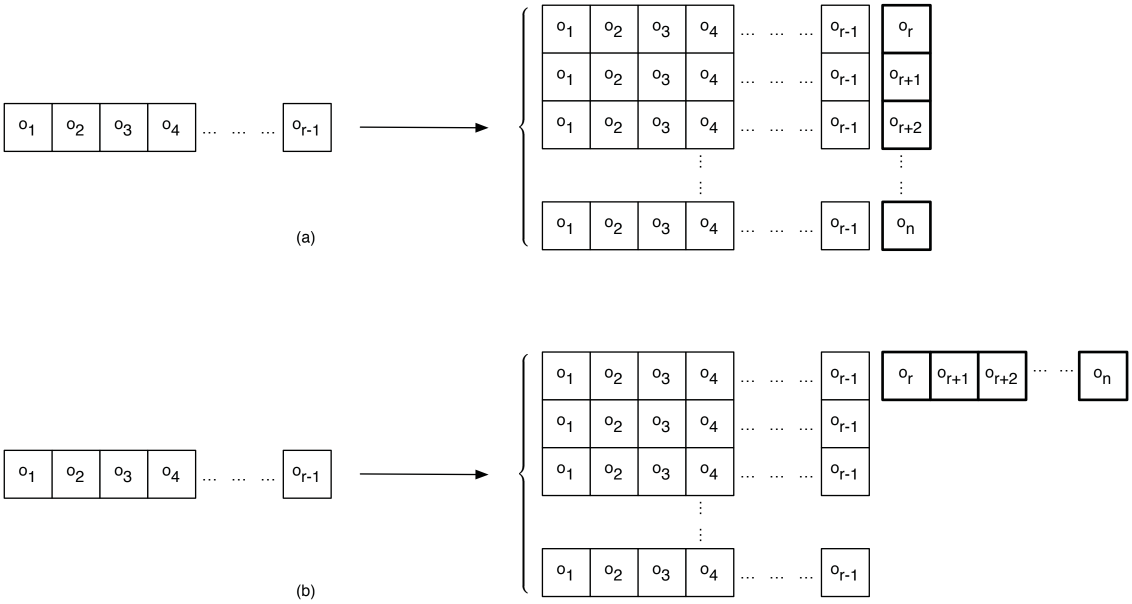


Figure 7.5: These are equivalent ways of viewing the number of members in an r-permutation from a counting perspective.

With respect to the proposition, we need to determine the number of occurrences of each member of O in this set of r-permutations. The induction hypothesis says that each object in set O occurs $\text{Freq}(n, r-1)$ times in the set of r–1-permutations of O. Let us assume that this is true and then figure out how we can use it to compute the number of r-permutations and convince ourselves that the result is correct.

First, for purposes of counting, let us rearrange the sequences of objects in Figure 7.5(a) to look like what is depicted in Figure 7.5(b). This makes it easier for one to see that the number of occurrences for each of the n objects in an r-permutation is

$$\begin{aligned}
1 \times P(n, r-1) + (n-r)\,\text{Freq}(n, r-1) &= \frac{n\,\text{Freq}(n, r-1)}{r-1} + (n-r)\,\text{Freq}(n, r-1) \\
&= \frac{n\,\text{Freq}(n, r-1)}{r-1} + \frac{(r-1)(n-r)\,\text{Freq}(n, r-1)}{r-1} \\
&= \frac{(n + rn - r^2 - n + r)\,\text{Freq}(n, r-1)}{r-1} \\
&= \frac{(rn - r^2 + r)\,\text{Freq}(n, r-1)}{r-1}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{r(n-r+1)\,\mathrm{Freq}(n, r-1)}{r-1} \\
&= \frac{r(n-r+1)}{r-1}\,\mathrm{Freq}(n, r-1) \\
&= \mathrm{Freq}(n, r) \\
&= \frac{rP(n, r)}{n}
\end{aligned}
$$

because if $\mathrm{Freq}(n, r) = \frac{rP(n,r)}{n}$ and $\mathrm{Freq}(n, r-1) = \frac{rP(n,r-1)}{n}$ for $n \geq r \geq 2$, then the original equation for $\mathrm{Freq}(n, r)$ can be rewritten as $\mathrm{Freq}(n, r) = \frac{r(n-r+1)}{r-1}\mathrm{Freq}(n, r-1)$. □

Theorem 7.2.2. *Each of the $n \geq r \geq 1$ distinct members in set $O = \{o_1, o_2, \ldots, o_n\}$ occurs $\frac{rC(n,r)}{n}$ times in the sample space of r-combinations for that set.*

Proof. This follows in a rather straightforward fashion from Theorem 7.2.1 on page 272. The key point to notice is that this theorem is concerned with combinations rather than permutations. Therefore, order does not matter and the $P(n, r)$ permutations of n distinct objects taken r at a time are all equivalent to each other. Hence, $C(n, r) = \frac{P(n,r)}{r!}$ (Rosen et al., 2000). This means that the number of times that each distinct member in set O occurs in the sample space of r-combinations is

$$
\begin{aligned}
\frac{\frac{r\,P(n,r)}{n}}{r!} &= \frac{r\,P(n, r)}{n\,r!} \\
&= \frac{r}{n}\frac{P(n, r)}{r!} \\
&= \frac{r}{n}\,C(n, r) \\
&= \frac{r\,C(n, r)}{n}.
\end{aligned}
$$

□

7.2.3 Compute the Average Unnormalized Position of a Relevant Document from a Sample Space of Orderings

To compute the mean unnormalized position of a relevant document in the sample space for a collection of N documents, given that we know the values of r_1, r_0, s_1, and s_0 for that collection, we need to determine the sum of the numbers that correspond to the positions associated with each relevant document of a sample point in the sample space. One way of doing this is, for each sample point, sum up the positions associated with the relevant documents, and, after calculating these values, compute the grand total from these sample point-specific sums. How can we use analytic techniques to do the equivalent of this?

One approach is to take a small case (i.e., a collection of a few documents), construct the sample space for it, and then study that in the hope of observing some useful insights, patterns, or relationships that can be used to help develop an analytical solution. Of course, any conjecture(s) that emanate from this study must be rigorously proved before they can be used in the solution.

Determining the Sample Space

In an optimal ordering, the positions of the n_1 documents with feature frequency 1 are in the closed interval $[1, n_1]$ (shown by Figure 7.1 on page 265) because, by definition, all the documents with that feature frequency (there are n_1 of them) are at the front of the ordering (and hence appear before any document with feature frequency 0). But, in a non-optimal ordering, these documents are not guaranteed to be constrained to that interval. In the latter situation, the positions of those n_1 documents (and likewise r_1 relevant documents with feature frequency 1) can be anywhere in the closed interval $[1, N]$. However, because the formula for $\mathcal{A}$ (Losee, 1998) (shown by Equation 7.1.1 on page 263), also by definition, is with respect to an optimal ordering, the calculations below implicitly assume that the document ordering is optimal.

In an optimal ordering of N documents, there are two groups of non-overlapping documents: those n_1 at the front with feature frequency 1 and those n_0 at the back with feature frequency 0. The documents in each of these groups can be arranged in any order independent of those in the other group.

The sample space for the documents with feature frequency 1 has a total of $C(n_1, r_1)$ sample points because that is the number of ways that r_1 positions can be chosen out of n_1 distinct positions when it is irrelevant which one is chosen first, second, third, etc. Similarly, the sample space for the documents with feature frequency 0 has a total of $C(n_0, r_0)$ sample points. Due to the independence mentioned above, the joint sample space for these two groups is the Cartesian product of these groups and contains $C(n_1, r_1) \times C(n_0, r_0)$ sample points.

Calculations

Since ASL, the average search length, is synonymous with the average unnormalized position of a relevant document, we have

$$\begin{aligned}
\text{ASL} &= \frac{\text{sum of the positions occupied by the relevant documents}}{\text{number of positions occupied by the relevant documents}} \\
&= \frac{S_{\text{rel}}}{N_{\text{rel}}} \\
&= \frac{S_{\text{rel},1} + S_{\text{rel},0}}{N_{\text{rel},1} + N_{\text{rel},0}}.
\end{aligned}$$

The equations for the variables $S_{\text{rel},1}, S_{\text{rel},0}, N_{\text{rel},1}, N_{\text{rel},0}$ and the values they denote appear below.

$$\begin{aligned}
S_{\text{rel},1} &= \text{sum of the positions occupied by the relevant documents with } d = 1 \\
&= \frac{r_1 \binom{n_1}{r_1}}{n_1} \binom{n_0}{r_0} \sum_{i=1}^{n_1} i
\end{aligned}$$

$$= \frac{r_1 \binom{n_1}{r_1}}{n_1} \binom{n_0}{r_0} \binom{n_1+1}{2}. \tag{7.2.1}$$

$$\begin{aligned} S_{\text{rel},0} &= \text{sum of the positions occupied by the relevant documents with } d = 0 \\ &= \frac{r_0 \binom{n_0}{r_0}}{n_0} \binom{n_1}{r_1} \sum_{i=n_1+1}^{N} i \\ &= \frac{r_0 \binom{n_0}{r_0}}{n_0} \binom{n_1}{r_1} \left[\binom{N+1}{2} - \binom{n_1+1}{2} \right]. \end{aligned} \tag{7.2.2}$$

$$\begin{aligned} N_{\text{rel},1} &= \text{number of the positions occupied by the relevant documents with } d = 1 \\ &= \left[r_1 \binom{n_1}{r_1} \right] \binom{n_0}{r_0}. \end{aligned} \tag{7.2.3}$$

$$\begin{aligned} N_{\text{rel},0} &= \text{number of the positions occupied by the relevant documents with } d = 0 \\ &= \left[r_0 \binom{n_0}{r_0} \right] \binom{n_1}{r_1}. \end{aligned} \tag{7.2.4}$$

The fraction in Equation 7.2.1 represents the number of times that each of the positions in the closed interval $[1, n_1]$ appears in the sample space and is occupied by a relevant document. The binomial expression in this equation represents how many combinations of relevant document positions in the closed interval $[n_1 + 1, N]$ are associated with each document combination of relevant document positions in the closed interval $[1, n_1]$. The summation represents the addition of the positions for those documents with feature frequency 1.

The fraction in Equation 7.2.2 represents the number of times that each of the positions in the closed interval $[n_1 + 1, N]$ appears in the sample space and is occupied by a relevant document. The binomial expression in this equation represents how many combinations of relevant document positions in the closed interval $[1, n_1]$ is associated with each document combination of relevant document positions in the closed interval

$[n_1+1, N]$. The summation represents the addition of the positions for those documents with feature frequency 0.

The bracketed term in Equation 7.2.3 on the preceding page represents the number of positions occupied by relevant documents in the sample space for relevant documents with feature frequency 1. Since each sample point in that space is associated with $\binom{n_0}{r_0}$ sample points in the sample space for documents with feature frequency 0, the total number of sample points in the joint sample space is the product of those values.

The bracketed term in Equation 7.2.4 on the previous page represents the number of positions occupied by relevant documents in the sample space for relevant documents with feature frequency 0. Since each sample point in that space is associated with $\binom{n_1}{r_1}$ sample points in the sample space for documents with feature frequency 1, the total number of sample points in the joint sample space is the product of those values.

The equations above can be simplified further. How to do that is demonstrated below.

$$\begin{aligned}
S_{\mathrm{rel}} &= S_{\mathrm{rel},1} + S_{\mathrm{rel},0} \\
&= \frac{r_1\binom{n_1}{r_1}}{n_1}\binom{n_0}{r_0}\binom{n_1+1}{2} + \frac{r_0\binom{n_0}{r_0}}{n_0}\binom{n_1}{r_1}\left[\binom{N+1}{2} - \binom{n_1+1}{2}\right] \\
&= \binom{n_1}{r_1}\binom{n_0}{r_0}\left[\frac{r_1}{n_1}\binom{n_1+1}{2} + \frac{r_0}{n_0}\left[\binom{N+1}{2} - \binom{n_1+1}{2}\right]\right]. \\
N_{\mathrm{rel}} &= \left[r_1\binom{n_1}{r_1}\right]\binom{n_0}{r_0} + \left[r_0\binom{n_0}{r_0}\right]\binom{n_1}{r_1} \\
&= (r_1+r_0)\binom{n_1}{r_1}\binom{n_0}{r_0}.
\end{aligned}$$

Putting all of this together, we obtain

$$\begin{aligned}
\mathrm{ASL} &= \frac{S_{\mathrm{rel}}}{N_{\mathrm{rel}}} \\
&= (r_0+r_1)^{-1}\left(\frac{r_1}{n_1}\binom{n_1+1}{2} + \frac{r_0}{n_0}\left[\binom{N+1}{2} - \binom{n_1+1}{2}\right]\right).
\end{aligned}$$

Now, if we expand the binomial terms, replace N by $n_1 + n_0$, and do some minor simplification, we obtain

$$\begin{aligned} \text{ASL} &= (r_0 + r_1)^{-1} 2^{-1} \left(\frac{r_1}{n_1}(n_1+1)n_1 + \frac{r_0}{n_0}\left[(n_1+n_0+1)(n_1+n_0) - (n_1+1)n_1\right] \right) \\ &= (r_0 + r_1)^{-1} 2^{-1} \left(r_1(n_1+1) + \frac{r_0}{n_0}\left[(n_1+n_0+1)(n_1+n_0) - (n_1+1)n_1\right] \right). \end{aligned} \tag{7.2.5}$$

For the moment, we concentrate on simplifying the part of the prior equation that is represented by

$$\frac{r_0}{n_0}\left[(n_1+n_0+1)(n_1+n_0) - (n_1+1)n_1\right]. \tag{7.2.6}$$

After this simplification has been accomplished, we plug that result in our immediately prior equation for ASL and proceed to derive the final version of this equation. After multiplying the parenthesized expressions, we obtain

$$\frac{r_0}{n_0}\left[n_1^2 + n_1 n_0 + n_0 n_1 + n_0^2 + n_1 + n_0 - n_1^2 - n_1\right].$$

After simplification, mainly by eliminating the terms that cancel each other, we have

$$\frac{r_0}{n_0}\left[n_1 n_0 + n_0 n_1 + n_0^2 + n_0\right] = r_0\left[n_1 + n_1 + n_0 + 1\right]. \tag{7.2.7}$$

The final step of deriving a simplified equation for ASL consists of substituting the expression on the right hand side of Equation 7.2.7 for the part of Equation 7.2.5 that is represented by Expression 7.2.6. This substitution yields

$$\begin{aligned} \text{ASL} &= (r_0 + r_1)^{-1} 2^{-1} \left(r_1(n_1+1) + r_0\left[n_1 + n_1 + n_0 + 1\right] \right) \\ &= (r_0 + r_1)^{-1} 2^{-1} \left(r_1 n_1 + r_1 + 2 r_0 n_1 + r_0 n_0 + r_0 \right) \end{aligned}$$

$$
\begin{aligned}
&= (r_0 + r_1)^{-1} 2^{-1} \left(r_1 + r_0 + (r_1 + r_0)n_1 + (n_0 + n_1)r_0\right) \\
&= (r_0 + r_1)^{-1} 2^{-1} \left((r_1 + r_0)(n_1 + 1) + (n_0 + n_1)r_0\right) \\
&= \frac{n_1 + 1}{2} + \frac{r_0 N}{2R} \\
&= \frac{R(n_1 + 1) + r_0 N}{2R}. \qquad (7.2.8)
\end{aligned}
$$

7.2.4 Derivation of the Formula for $\mathcal{A}$

The formula for $\mathcal{A}$ can be derived in an indirect way by computing the ASL and then rewriting the formula so that it fits the template below. Without loss of generality, if we assume optimal ranking (i.e., $Q = 1$), then ASL $= N\mathcal{A} + \frac{1}{2}$ (the template). The ASL is simply the unnormalized average of the positions occupied by the relevant documents in an ordering over all the possible orderings for a collection with $N = r_0 + r_1 + s_0 + s_1$.

For the convenience of the reader, we restate below, from Equation 7.2.8, that

$$
\text{ASL} = \frac{R(n_1 + 1) + r_0 N}{2R},
$$

assuming, of course, that the number of relevant documents is at least 1.

After rewriting, we obtain

$$
\begin{aligned}
\text{ASL} &= \frac{n_1 + 1}{2} + \frac{r_0 N}{2(r_1 + r_0)} \\
&= \frac{n_1}{2} + \frac{r_0 N}{2(r_1 + r_0)} + \frac{1}{2} \\
&= \frac{n_1 N}{2N} + \frac{r_0 N}{2(r_1 + r_0)} + \frac{1}{2} \\
&= N\left[\frac{n_1}{2N} + \frac{r_0}{2(r_1 + r_0)}\right] + \frac{1}{2}. \qquad (7.2.9)
\end{aligned}
$$

Therefore,

$$\begin{aligned}\mathcal{A} &= \frac{n_1}{2N} + \frac{r_0}{2(r_1 + r_0)} \\ &= \frac{n_1}{2N} + \frac{r_0}{2R} \\ &= \frac{n_1 R + r_0 N}{2NR}. \qquad (7.2.10)\end{aligned}$$

Figure 7.6 on page 285 contains histograms of the distributions of $\mathcal{A}$ values when $N =$ 10, 20, and 50. Note how the histograms become more symmetrical as the value of N increases.

Lemma 7.2.3. *The probabilistic and combinatoric formulas for $\mathcal{A}$ are equivalent.*

Proof. Since,

$$\begin{aligned}\Pr(d|\mathit{rel}) &= \frac{\Pr(d, \mathit{rel})}{\Pr(\mathit{rel})} \\ &= \frac{\text{\# of relevant documents with ff 1}}{\text{total \# of relevant documents}} \\ &= \frac{r_1}{R}\end{aligned}$$

and $\Pr(d) = n_1/N$, then $\mathcal{A} = (1 + t - p)/2$ can be expressed as

$$\begin{aligned}\mathcal{A}_{\mathit{probabilistic}} &= \left(1 + \frac{n_1}{N} - \frac{r_1}{R}\right)/2 \\ &= (NR + n_1 R - N r_1)/(2NR). \qquad (7.2.11)\end{aligned}$$

Similarly, Equation 7.2.10, that is, $\mathcal{A} = n_1/(2N) + r_0/(2R)$, can be expressed as

$$\begin{aligned}\mathcal{A}_{\mathit{combinatoric}} &= \frac{n_1}{2N} + \frac{r_0}{2R} \\ &= \frac{n_1 R}{2NR} + \frac{r_0 N}{2RN}\end{aligned}$$

$$= \frac{n_1 R + r_0 N}{2NR}. \tag{7.2.12}$$

Since $\mathcal{A}_{probabilistic}$ and $\mathcal{A}_{combinatoric}$ now have the same denominators, it suffices to just show that the numerators are equivalent. Below, the notation lhs $\overset{?}{\equiv}$ rhs denotes a situation where the expression on the left-hand side (*lhs*) of the $\overset{?}{\equiv}$ symbol might not be equivalent to the expression on the right hand side side (*rhs*) of that symbol.

Equation 7.2.13 asks if the numerators on the final lines of Equations 7.2.11 and 7.2.12 are equivalent. Since term $n_1 R$ appears once in both numerators, this comparison simplifies to Equation 7.2.14. After factoring out N on the left hand side of the comparison and permuting the terms on its right hand side, we obtain Equation 7.2.15. Since $R = r_1 + r_0$, this comparison can be rewritten as Equation 7.2.16. Obviously, at this point, we can say that the original numerators were equivalent expressions because Nr_0 equals Nr_0.

$$\begin{aligned} NR + n_1 R - Nr_1 &\overset{?}{\equiv} n_1 R + r_0 N \qquad (7.2.13) \\ NR - Nr_1 &\overset{?}{\equiv} r_0 N \qquad (7.2.14) \\ N(R - r_1) &\overset{?}{\equiv} Nr_0 \qquad (7.2.15) \\ Nr_0 &\overset{?}{\equiv} Nr_0 \qquad (7.2.16) \end{aligned}$$

Hence, Equation 7.2.11 and Equation 7.2.12 are equivalent, thus allowing us to state that

$$\begin{aligned} \mathcal{A}_{combinatoric} &= \mathcal{A}_{probabilisticic} \\ &= \frac{n_1 R + r_0 N}{2NR}. \end{aligned} \tag{7.2.17}$$

□

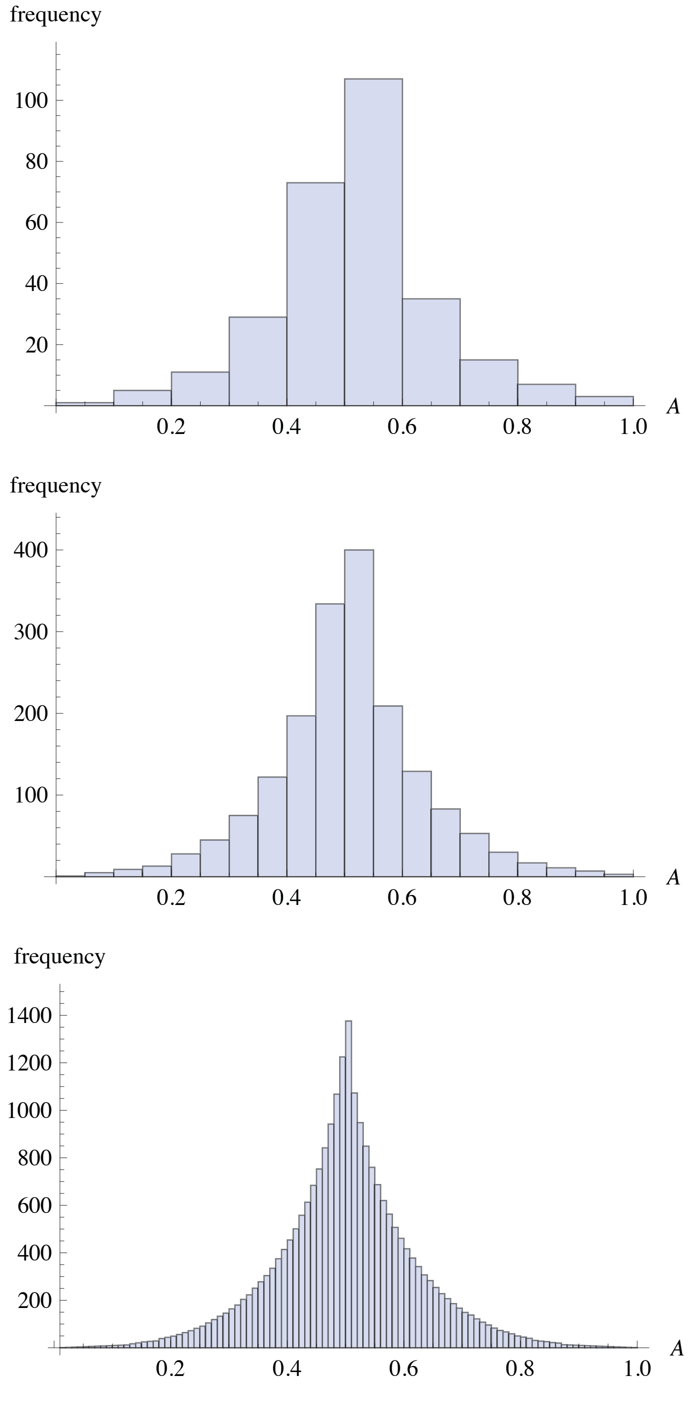


Figure 7.6: Distributions of $\mathcal{A}$ values for $N = 10,\ 20,$ and $50,$ respectively.

7.3 Gaussian Polynomials and Some of Their Properties

In this dissertation, Gaussian polynomials (Andrews, 1984; Andrews and Eriksson, 2004; Goulden and Jackson, 1983) are of use for two primary reasons: their ability to help model the distributions of sums of document positions in an optimal ranking and, later, in Chapter 8, their use in the development of an improved formula for computing the ASL. Each of these reasons involve finding all the sums of k-subsets of sets of n positive integers such that $k \leq n$ and both k and n are natural numbers.

Definition 7.3.0.1. The q-binomial coefficient (also known as Gaussian coefficient or Gaussian polynomial) is

$$\begin{bmatrix} n \\ m \end{bmatrix}_q = \prod_{i=1}^{m} \frac{1-q^{n-m+i}}{1-q^i}$$

for $n, m \in \mathbb{N}$; where $\mathbb{N}$ denotes the set of natural numbers.

Theorem 7.3.1. *(Andrews, 1984) Let $0 \leq m \leq n$ be integers. The Gaussian polynomial $\begin{bmatrix} n \\ m \end{bmatrix}_q$ is a polynomial of degree m (n - m) in q that satisfies the following relations.*

$$\begin{aligned}
\begin{bmatrix} n \\ 0 \end{bmatrix}_q &= \begin{bmatrix} n \\ n \end{bmatrix}_q = 1 \\
\begin{bmatrix} n \\ m \end{bmatrix}_q &= \begin{bmatrix} n \\ n-m \end{bmatrix}_q \\
\begin{bmatrix} n \\ m \end{bmatrix}_q &= \begin{bmatrix} n-1 \\ m \end{bmatrix}_q + q^{n-m} \begin{bmatrix} n-1 \\ m-1 \end{bmatrix}_q \\
\begin{bmatrix} n \\ m \end{bmatrix}_q &= \begin{bmatrix} n-1 \\ m-1 \end{bmatrix}_q + q^{m} \begin{bmatrix} n-1 \\ m \end{bmatrix}_q \\
\lim_{q \to 1} \begin{bmatrix} n \\ m \end{bmatrix}_q &= \frac{n!}{m!(n-m)!} = \binom{n}{m}
\end{aligned}$$

Proof. The proof of this theorem can be found in Andrews (1984). □

7.3.1 A Motivating Example: The Use of Gaussian Polynomials to Obtain Document Position Distributional Information

Let $P = \{1, 2, 3, 4, 5\}$ be a set of 5 document positions. Assume that there are 3 relevant documents and that it is equally likely that a relevant document can occupy any one of these positions. Furthermore, a position can be occupied by, at most, one document. Hence, $n = 5$ and $k = 3$. By the Binomial Theorem, there are

$$\binom{n}{k} = \binom{5}{3} = 10$$

3-subsets of these positions. These subsets are

$$\{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,3,4\}, \{1,3,5\}, \{1,4,5\}, \{2,3,4\}, \{2,3,5\}, \{2,4,5\}, \text{ and } \{3,4,5\}.$$

The sums of the positions that correspond to these 3-subsets are

$$6, 7, 8, 8, 9, 10, 9, 10, 11, \text{ and } 12,$$

respectively. Note that in this sums-of-positions distribution the values of $8, 9$, and 10 occur with frequency 2 whereas the four remaining values, that is, $6, 7, 11$, and 12, each occur with a frequency of 1. If the sums are ascendingly ordered, the value sequence is

$$6, 7, 8, 8, 9, 9, 10, 10, 11, 12.$$

Among the 10 values, only 7 are unique. Gaussian polynomials can be used to determine this distribution. By Definition 7.3.0.1 on the previous page, the Gaussian polynomial for the given values of n and k is

$$\begin{aligned}
\begin{bmatrix}5\\3\end{bmatrix}_q &= \prod_{i=1}^{3} \frac{1-q^{5-3+i}}{1-q^i} \\
&= \frac{(1-q^3)(1-q^4)(1-q^5)}{(1-q)(1-q^2)(1-q^3)} \\
&= \frac{(1-q^4)(1-q^5)}{(1-q)(1-q^2)} \\
&= \frac{(1-q^2)(1+q^2)(1-q^5)}{(1-q)(1-q^2)} \\
&= \frac{(1+q^2)(1-q^5)}{(1-q)} \\
&= (1+q^2)(1+q+q^2+q^3+q^4) \\
&= 1+q+2q^2+2q^3+2q^4+q^5+q^6.
\end{aligned}$$

Since the lowest value sum is $1 + 2 + 3 = 6$, we need to adjust the previous equation by that information in order to obtain the distributional information for the values that were used in this example. This means that we now have

$$\begin{aligned}
q^6\begin{bmatrix}5\\3\end{bmatrix}_q &= q^6(1+q+2q^2+2q^3+2q^4+q^5+q^6) \\
&= q^6+q^7+2q^8+2q^9+2q^{10}+q^{11}+q^{12}.
\end{aligned}$$

From it, we obtain the following distributional information: the lowest-valued sum (i.e., 6) occurs once, the next lowest-value sum (i.e., 7) occurs once, the third lowest-valued one (i.e., 8) occurs twice, and so on, with the highest-valued sum (i.e., 12) occurring one time.

7.3.2 Reciprocity and Unimodality

In addition to the properties listed in Theorem 7.3.1 on page 286, the Gaussian polynomials (or q-binomial coefficients) have other similar properties to regular binomial coefficients largely due to their being the q-analogs of these entities. Two of these other properties that are of interest in this dissertation is that the distributions associated with q-binomial coefficients are both *reciprocal* (i.e., symmetrical) and *unimodal* (i.e., monotonic on both sides of the midpoint) .

> DEFINITION 3.6. A polynomial $p(q) = a_0 + a_1 q + \cdots + a_n q_n$ is called *reciprocal* if for each i, $a_i = a_{n-i}$, equivalently $q^n p(q^{-1}) = p(q)$.
>
> DEFINITION 3.7. A polynomial $p(q) = a_0 + a_1 q + \cdots + a_n q_n$ is called *unimodal* if there exists m such that
>
> $$a_0 \leq a_1 \leq a_2 \leq \cdots \leq a_m \geq a_{m+1} \geq a_{m+2} \geq \cdots a_n.$$
>
> ...
>
> THEOREM 3.9. Let $p(q)$ and $r(q)$ be reciprocal, unimodal polynomials with nonnegative coefficients; then $p(q)r(q)$ is also a reciprocal, unimodal polynomial with nonnegative coefficients. (Andrews, 1984)

The *reciprocal* property means that, for a polynomial $p(q)$, if one knows the value v of coefficient a_i, for $0 \leq i \leq n$, then one does not need to calculate the value of a_{n-i} because its value is also this same value v. A practical aspect of this is that the effort to calculate these coefficients can be cut approximately in half. That is, one needs to only calculate the values for the coefficients with indices 0, 1, ..., $\lfloor n/2 \rfloor$, inclusive, then use the reciprocal property to calculate the values that correspond to indices $\lfloor n/2 \rfloor + 1$, $\lfloor n/2 \rfloor + 2$, ..., n, inclusive. For example, if $n = 5$ and $a_2 = v$, then, from the reciprocal property, we do not need to calculate the value for $a_{5-2} = a_3$ because we know that it must be the same as v, the value for a_2. Many famous distributions (e.g., the normal distribution (a continuous distribution), the binomial distribution (a discrete distribution) are reciprocal (i.e., symmetrical). From a statistical viewpoint, one of the qualities of a reciprocal distribution is that its mean, median, and mode are the same value.

The *unimodal* property means that the series of a_i values first increases up to a point and then decreases, if the a_0 value is different than the $a_{\lfloor n/2 \rfloor}$ (i.e., middle) value. Another way to view this is that the value at the beginning (and ending) of the sequence is the same as the minimum value of the sequence and that the middle value is the same as the maximum value of the sequence.

The theorem from Andrews (1984) states that the convolution of two polynomials that are both reciprocal and unimodal, with nonnegative coefficients, yields a new polynomial that is also reciprocal, unimodal, and has no nonnegative coefficients. This means that when we combine two polynomials that are both unimodal and reciprocal, we are rewarded with a mixture polynomial that is also unimodal and reciprocal.

The examples in Section 7.5 illustrate the effects of the reciprocal and unimodal properties. There, in that section, we make use of these properties to determine the distribution of the sums of the positions of the relevant documents in an optimal ranking.

7.3.3 Additional Important Relationships

The concepts of *constrained parts*, *constrained multiplicities of parts*, *convolution*, and the *Cauchy Binomial Theorem* (both plain and extended forms) are very important in the derivations of the equations for normalized and unnormalized search length. These concepts are discussed below.

Definition 7.3.3.1. Partitions with constrained parts and constrained multiplicities of parts.

> Let two sets W, of nonnegative integers, and R, of positive integers, be given, with $0 \in W$. Let $p(n, k; W, R)$ be the number of partitions of n into k parts such that all of the parts lie in R, and all of their multiplicities lie in W. Then ...
>
> $$\sum_{n,k} p(n, k; W, R) x^n y^k = \prod_{r \in R} \left(\sum_{k \in W} y^k x^{kr} \right).$$
>
> From this generating function we can prove many theorems about partitions. (Wilf, 2006)

When $R \in \{1, 2, \ldots, n\}$ and $W \in \{0, 1\}$, we have the special case

$$\begin{aligned}\sum_{n,k} p(n, k; \{0, 1\}, \{1, 2, \ldots, n\}) x^n y^k &= \prod_{r=1}^{n} \left(\sum_{k=0}^{1} y^k x^{kr} \right) \\ &= \prod_{r=1}^{n} (1 + yx^r).\end{aligned}$$

This is the ordinary generating function for determining all the possible sums (and their frequency counts) when k values, without replacement, are selected from the set of integers that range from 1 to n, inclusive. This is a very well-known generating function in the area of combinatorics known as integer partition theory. It is known as the Cauchy binomial theorem and is formally stated below as a q-series.

Theorem 7.3.2. *Cauchy Binomial Theorem*

$$\prod_{i=1}^{n} (1 + zq^i) = \sum_{m=0}^{\infty} q^{\binom{m+1}{2}} \begin{bmatrix} n \\ m \end{bmatrix}_q z^m.$$

Proof. Gasper and Rahman (2004) contains a proof of this theorem. Their notation, though, is different than the notation that was used in the above equation. □

The Cauchy binomial theorem is used to model the total search lengths for the feature frequency 1 part of an ordering. Unless this part of an ordering has zero documents, the theorem, as stated, cannot be used to model the feature frequency 0 part of an ordering because it assumes that the minimum element in R is 1. However, we can easily extend the theorem so that it handles *any* sequence of distinct positive integers in a closed interval $[1 + s, n + s]$ where $s \in \mathbb{N}$, that is, s is a natural number that represents how many positions the elements in a sequence are to be *shifted* from lower-indexed positions to higher-indexed positions.

Theorem 7.3.3. *Cauchy Binomial Theorem (extended)*

$$\prod_{i=1+s}^{n+s}(1+zq^i) = \sum_{m=0}^{\infty} q^{\binom{m+1}{2}+ms} \begin{bmatrix} n \\ m \end{bmatrix}_q z^m$$

Proof. Since $\begin{bmatrix} n \\ m \end{bmatrix}_q = 0$, when $m > n$, and $\binom{m+1}{2}$ can be rewritten as $m(m+1)/2$,

$$\prod_{i=1+s}^{n+s}(1+zq^i) = \sum_{m=0}^{n} q^{m(m+1)/2+ms} \begin{bmatrix} n \\ m \end{bmatrix}_q z^m.$$

Basis step. n=1.

$$\begin{aligned} 1 + q^{1+s}z &= q^{0+0} \begin{bmatrix} 1 \\ 0 \end{bmatrix}_q z^0 + q^{1+s} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_q z^1 \\ &= 1 + q^{1+s}z. \end{aligned}$$

Inductive step. Assume that the induction hypothesis is true for $0 \le m \le n-1$. For the inductive step, we must expand

$$\left(\sum_{m=0}^{n-1} q^{m(m+1)/2+ms} \begin{bmatrix} n-1 \\ m \end{bmatrix}_q z^m \right) \left(1 + q^{n+s}z\right)$$

and extract from it the coefficient of z^m.

This coefficient is

$$\begin{aligned} & q^{m(m+1)/2+ms} \begin{bmatrix} n-1 \\ m \end{bmatrix}_q + q^{(m-1)m/2+(m-1)s+n+s} \begin{bmatrix} n-1 \\ m-1 \end{bmatrix}_q \\ = \;& q^{m(m+1)/2+ms} \left(\begin{bmatrix} n-1 \\ m \end{bmatrix}_q + q^{n-m} \begin{bmatrix} n-1 \\ m-1 \end{bmatrix}_q \right) \\ = \;& q^{\binom{m+1}{2}+ms} \begin{bmatrix} n \\ m \end{bmatrix}_q . \end{aligned}$$

A generalized form of the binomial theorem (Larsen, 2007) can be used to develop an alternate proof of this theorem. □

The next lemma uses the results from the extended version of the Cauchy binomial theorem to derive an expression that describes the distribution of the total search lengths for an N document collection that has r_1 relevant documents that contain the query term, r_0 relevant documents that do not contain the query term, s_1 non-relevant documents that contain the query term, and s_0 non-relevant documents that do not contain the query term.

Theorem 7.3.4. *Let $r_0, r_1, s_0, s_1 \in \mathbb{N}$ represent the parameters of an N-document collection for $N = n_0 + n_1$ with $n_1 = s_1 + r_1$ and $n_0 = s_0 + r_0$. The distribution of total search lengths for a collection with these parameters is described by*

$$q^{\binom{r_0+1}{2}+r_0\cdot n_1+\binom{r_1+1}{2}} \begin{bmatrix} n_0 \\ r_0 \end{bmatrix}_q \begin{bmatrix} n_1 \\ r_1 \end{bmatrix}_q .$$

Proof. By the extended version of the Cauchy binomial theorem, the expression that describes the total search length distribution for the feature frequency 0 part of an ordering is

$$q^{\binom{r_0+1}{2}+r_0\cdot n_1} \begin{bmatrix} n_0 \\ r_0 \end{bmatrix}_q ,$$

when $n = n_0$ and $m = r_0$. Also, by this theorem, the analogous expression for the distribution for the feature frequency 1 part of an ordering is

$$q^{\binom{r_1+1}{2}} \begin{bmatrix} n_1 \\ r_1 \end{bmatrix}_q ,$$

when $n = n_1$ and $m = r_1$.

The expression for the *convolution* (combined distribution) of total search lengths is the product of the expressions for the individual expressions:

$$\begin{aligned}
& \left(q^{\binom{r_0+1}{2}+r_0\cdot n_1} \begin{bmatrix} n_0 \\ r_0 \end{bmatrix}_q \right) \left((q^{\binom{r_1+1}{2}} \begin{bmatrix} n_1 \\ r_1 \end{bmatrix}_q \right) \\
= \quad & q^{\binom{r_0+1}{2}+r_0\cdot n_1+\binom{r_1+1}{2}} \begin{bmatrix} n_0 \\ r_0 \end{bmatrix}_q \begin{bmatrix} n_1 \\ r_1 \end{bmatrix}_q .
\end{aligned}$$

This result is used in the next section to help develop generating functions that describe the distribution of total search lengths. □

7.3.4 Performance Evaluation Implications for Information Retrieval Research

Gaussian polynomials can be used to help construct combinatoric, probabilistic, and mathematical models of information retrieval performance measures that sequence the documents in a collection, with respect to a query q, according to retrieval status values that are based on nondichotomous (e.g., degrees of relevance, graded relevance assessments) (Cuadra and Katter, 1967; Spink et al., 1998; Tang et al., 1999; Vakkari and Hakala, 2000; Kekäläinen and Järvelin, 2002), rather than on binary relevance assessments. It is assumed that the set of possible nondichotomous relevance assessments is finite (i.e., a fixed number of categories), rather than infinite (i.e., continuous relevance), has a cardinality of at least three, and that the cardinality is moderate in value (e.g., five to ten assessment categories).

There are two major parameters that can be varied with respect to the assessments: (1) the number of categories and (2) the relative weight of one category to another. For example, suppose the categories are the same as those used for the Cystic Fibrosis test collection (i.e., highly relevant, marginally relevant, not relevant). There is nothing sacrosanct about these assessments as some studies, such as those in this dissertation, use less categories (i.e., binary relevance judgments) whereas others use more categories

(Cuadra and Katter, 1967; Spink et al., 1998; Tang et al., 1999; Vakkari and Hakala, 2000; Kekäläinen and Järvelin, 2002).

An IR researcher may want to undertake a study where the focus is on investigating how performance measures are affected as a function of the number of assessment categories. Another factor that can be studied independently, if desired, is the effect that different category weights have on the rankings and performance measure evaluations. For example, the three relevance judgment categories used in the CF test collection could be given weights of 0, 1, and 2, respectively. In essence, this says that a non-relevant document has no value and that a highly relevant one has a weight that is twice that of a marginally relevant one. A researcher might want to, say, study the effect of changing the weight for a marginally relevant document to 2 and that of a highly relevant document to 5. One might say here that the the highly relevant document now has a weight that is larger relative to the marginally relevant document than it did prior to the change.

The above paragraph enumerated examples of some of the types of IR performance evaluation studies that could be facilitated by the use of Gaussian polynomials for the study of a performance measure such as the Average Search Length. However, there is nothing that limits its use to the ASL. It could be adapted, with varying amounts of ease, to help model distributions of ranked documents for other performance measures.

7.4 Probability Mass Functions, Generating Functions, and Probability Generating Functions

A *probability mass function* (pmf) defines the probability distribution of a discrete random variable, whereas a *probability density function* (pdf) defines the probability distribution of a continuous random variable. A discrete random variable can only assume integral (i.e., integer) values within an interval, or over several intervals. Similarly, a continuous random variable can only assume real values. Since the work in this chapter (and dissertation) only uses discrete random variables, we do not discuss probability density functions.

A *frequency distribution* (as contrasted with a probability distribution) specifies the frequency of each value of a discrete random variable. If this distribution is known, then it is simple to derive the pmf from it. All one needs to do is to divide each frequency by the number of events that are in the sample space for the random variable of interest. Figure 7.7 on page 298 has examples of both of these types of distribution.

A discrete random variable can only assume a finite (though possibly large) number of values. A discrete probability distribution is often given in the form of a table, a set of formulas, or a bar chart. The cumulative probability, across the range of values for a discrete distribution, is always 1. This probability is obtained by summing the associated probabilities for all values in the range. The summing technique is discrete summation.

A one-variable *generating function* f is a formal power series in a variable, say, x, whose coefficients succinctly encode information about a sequence a_i that is indexed by the natural numbers, i.e.,

$$\begin{aligned} f(x) &= \sum_{i=0}^{\infty} a_i x^i \\ &= a_0 + a_1 x + a^2 x^2 + a^3 x^3 + \cdots . \end{aligned}$$

The function $f(x)$ encodes information about the sequence of real values a_0, a_1, a_2, a_3, and so on. Generating functions are often employed by mathematicians, combinatorialists, and statisticians because of their parsimony in encoding information about sequences and also because of the well-developed theory of power series with non-negative coefficients.

A generating function can also be a function of several variables (like some of those that appear in the next section). A key way in which a formal power series differs from a power series is that we generally do not have to be concerned about whether a formal power series converges because we are typically only interested in it for the coefficients and the exponents that are associated with its terms.

As an example of how a generating function can succinctly encode information, let

$$f(x) = \frac{1}{1-2x}.$$

This generating function can be expanded to show the information that it encodes:

$$\begin{aligned} f(x) &= \frac{1}{1-2x} \\ &= a_0 + 2x + 2^2 x_2 + 2^3 x_3 + \cdots . \end{aligned}$$

We find that the coefficient 2^i is associated with each x_i, where i is a natural number.

The *probability generating function* (pgf) of a discrete random variable is a formal power series representation (i.e., the generating function) of the probability mass function of the random variable. It is of the form

$$\begin{aligned} f(x) &= \sum_{i=0}^{\infty} a_i x^i \\ &= p_0 + p_1 x + p_2 x^2 + p_3 x^3 + \cdots , \end{aligned}$$

where p_i represents the probability that the value of the random variable X is equal to the value of i, that is, $\Pr(X = i)$. For example, the pgf $f(x)$ that is associated with the data that Figure 7.7 is based on is

$$\begin{aligned} f(x) &= (x^7 + 2x^8 + 3x^9 + 3x^{10} + 3x^{11} + 2x^{12} + x^{13})/15 \\ &= (1/15)x^7 + (2/15)x^8 + (3/15)x^9 + (3/15)x^{10} + (3/15)x^{11} + (2/15)x^{12} + (1/15)x^{13} \end{aligned}$$

$$= (1/15)x^7 + (2/15)x^8 + (1/5)x^9 + (1/5)x^{10} + (1/5)x^{11} + (2/15)x^{12} + (1/15)x^{13}.$$

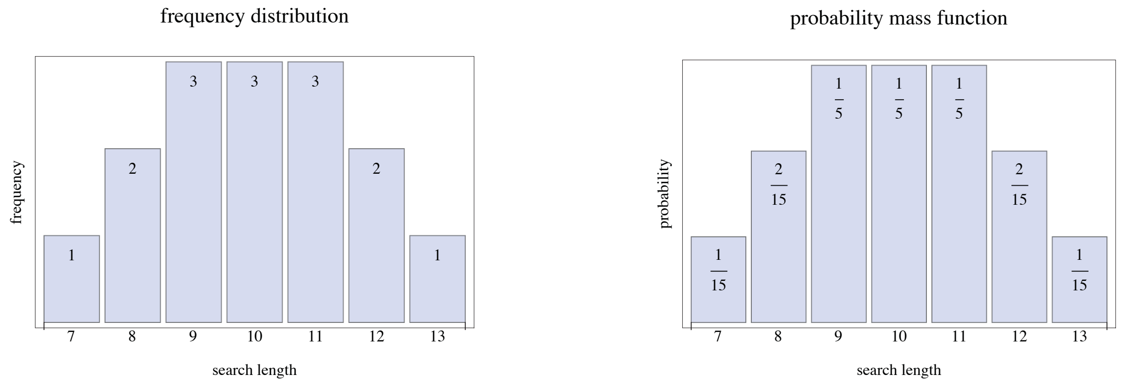


Figure 7.7: Bar charts for the frequency distribution and probability mass function of the data in Figure 7.2 on page 266. The leftmost part of this figure illustrates that there are seven distinct search lengths associated with the data. These lengths are 7, 8, 9, 10, 11, 12, and 13, respectively, with frequencies 1, 2, 3, 3, 3, 2, and 1. The rightmost part of this figure indicates that the probabilities that are associated with the seven distinct search lengths are 1/15, 2/15, 3/15, 3/15, 3/15, 2/15, and 1/15, respectively, because the sample space that is associated with this data has 15 events. In the graph for this probability mass function, the value 3/15 was simplified to 1/5.

7.5 The Distribution of the Sums of the Positions of the Relevant Documents in an Optimal Ranking

In an optimal ranking, all the documents that contain the query term (i.e., those with feature frequency 1) appear at the front of the ordering, whereas those that do not contain the term (i.e., those with feature feature 0) appear immediately after the last document in the former group of documents. The former group contains n_1 total documents, of which r_1 are relevant and $n_1 - r_1$, the remainder, are non-relevant. Similarly, the latter group contains n_0 total documents, of which r_0 are relevant and $n_0 - s_0$, the remainder, are non-relevant.

Assume that a document collection c of size $N = n_0 + n_1$, with $n_0 = r_0 + s_0$ and $n_1 = r_1 + s_1$, exists with $r_1, r_1, s_0, s_1 \in \mathbb{N}$. From the sample space determination discussion in Section 7.2.3, this collection of N documents has $C(n_0, r_0) \times C(n_1, r_1)$ possible distinct document orderings, with respect to query q. Let the notation $O(q, c)$ denote the set of document orderings for query q and collection c. Each of these distinct orderings has a total of $r_0 + r_1$ relevant documents and a total of $s_0 + s_1$ non-relevant documents. The *total search length for an individual ordering* (T_i), where $i \in O(q, c)$, is computed by finding the position of each of the relevant documents in this ordering i and, then, summing these values. The *mean search length for an individual ordering* ($\mathcal{M}_i$) is its T_i value divided by the number of relevant documents in that ordering.

The total search lengths (TSLs) and mean search lengths (MSLs) can be viewed as random variables. We are interested in studying the variance of the TSLs and MSLs for an $O(q, c)$ object that possesses the characteristics that were given in the previous paragraph. Later, we use these variances to help establish confidence intervals that aid us in our validation of the ASL. To compute these variances, we need to determine the TSL and MSL for each individual ordering and, then, compute the mean of these values over all of the orderings.

The effort starts with determining how the TSLs and MSLs are distributed for the N document collection c, and specified query q. The distribution determination process for an ordering consists of these three steps: find the distribution for the group of documents that have feature frequency 1, find the distribution for the group of documents that have feature frequency 0, and, then, combine the distributions to obtain the distribution for the entire ordering.

The combinatorial technique of generating functions (Graham et al., 1994; Charalambides, 2002; Lando, 2003; Wilf, 2006), in conjunction with Gaussian polynomials (Andrews, 1984; Andrews and Eriksson, 2004; Comtet, 1974), is probably the most direct way to accomplish the task of determining this distribution for arbitrary N, r_1, n_1, r_0, and n_0. A key to constructing the correct generating function G_0 is to recognize that the r_0 relevant document positions, selected from the part of the orderings corresponding to the documents where the term is absent, is equivalent to the process of selecting *without replacement* the r_0 documents from a population of n_0 documents that have positions in the closed interval $[n_1+1, N]$. The ordinary generating function G_0 for this part of the orderings is

$$G_0(x, y, n_1, N) = \prod_{i=n_1+1}^{N} (1 + x^i y). \tag{7.5.1}$$

The analogous ordinary generating function G_1 for the part of the ordering corresponding to the term being present is

$$G_1(x, z, n_1) = \prod_{i=1}^{n_1} (1 + x^i z). \tag{7.5.2}$$

A key to constructing G_1 is to recognize that the r_1 relevant document positions, selected from the part of the orderings corresponding to the documents where the term is present, is equivalent to the process of selecting *without replacement* the r_1 documents from a population of n_1 documents that have positions in the closed interval $[1, n_1]$.

The ordinary generating function for the entire ordering, $G(x, y, z, n_1, N)$, is simply the convolution (i.e., product) of the ordinary generating functions for the two parts of the ordering, namely, $G_0(x, y, n_1, N)$ and $G_1(x, z, n_1)$:

$$G(x, y, z, n_1, N) = G_0(x, y, n_1, N) \cdot G_1(x, z, n_1). \tag{7.5.3}$$

In order to determine the distribution of the TSLs and MSLs, we need to first expand the expression denoted by $G(x, y, z)$. After that, we need to extract the function of x that is the coefficient of the term $y^{r_0}z^{r_1}$. We use the function $T(x)$ to denote the TSL version of the expressions. The analogous expression for the MSL, $M(x)$, is easily derived from $T(x)$ as follows: divide each exponent by r_0+r_1, the number of relevant documents. The distribution information can be recovered as follows: the exponent of an x-term in $T(x)$ represents a total search length v and the coefficient of this term represents the number of orderings that had total search length v. Similarly, the exponent of an x-term in $M(x)$ represents a mean search length w and the coefficient of this term represents the number of orderings that had mean search length w. Basically, we now have the distribution information, that is, the values that occurred and the frequency for each one.

Definition 7.5.0.1. The *convolution* $C(x)$, of two ordinary generating functions $A(x)$ and $B(x)$, is $C(x) = A(x)B(x)$ if and only if $c_k = \sum\limits_{i=0}^{k} a_i b_{k-i}$ for $k \in \mathbb{N}$ where $\mathbb{N}$ denotes the set of natural numbers.

7.5.1 Another Motivating Example: The Use of Gaussian Polynomials and Probability Generating Functions to Obtain Search Length Means and Variances

This discussion about the distribution of sums is technical and somewhat lengthy. The running example is intended to facilitate understanding of its main concepts and is used to help illustrate the process that was just sketched out above. After the end of the example, there is a more formal treatment of the process and allied concepts. The data that is used comes from the scenario that is depicted by Figure 7.2 on page 266. The parameters for it are $N = n_1 + n_0 = 8$, $n_1 = r_1 + s_1 = 3$, $n_0 = r_0 + s_0 = 5$, with $r_1 = 2$, $s_1 = 1$, $r_0 = 1$, and $s_0 = 4$. More information on probability generating functions can be found in the material of Section 7.5.1.

Expanded versions of Equation 7.5.1 on the preceding page and Equation 7.5.2 on the previous page, the equations for $G_0(x, y, 3, 8)$ and $G_1(x, z, 3)$, respectively, appear below, with the appropriate value substitutions for parameters n_1 and N. These equations are purposely not in their simplest forms in order to make it easier for the reader to discern the relationships between the coefficients and exponents of the various terms.

$$\begin{aligned}
G_0(x,y,3,8) &= \prod_{i=3+1}^{8}(1+x^i y) \\
&= (1+x^4y)(1+x^5y)(1+x^6y)(1+x^7y)(1+x^8y) \\
&= (1)y^0 + \\
&\quad (x^4+x^5+x^6+x^7+x^8)y^1 + \\
&\quad (x^9+x^{10}+2x^{11}+2x^{12}+2x^{13}+x^{14}+x^{15})y^2 + \\
&\quad (x^{15}+x^{16}+2x^{17}+2x^{18}+2x^{19}+x^{20}+x^{21})y^3 + \\
&\quad (x^{22}+x^{23}+x^{24}+x^{25}+x^{26})y^4 + \\
&\quad (x^{30})y^5. \qquad (7.5.4)
\end{aligned}$$

The $(x^9+x^{10}+2x^{11}+2x^{12}+2x^{13}+x^{14}+x^{15})y^2$ term in the $G_0(x,y,3,8)$ equation has this interpretation: the number of distinct 2-addend sums that can be constructed from the set $\{4,5,6,7,8\}$ is 7; the sums range in value from 9 to 15, inclusive; and their respective frequencies are 1,1,2,2,2,1,1. This means that there was exactly one way to obtain the sum 9 (e.g. $4+5$); exactly one way to obtain the sum 10 (e.g., $3+7$); exactly two ways to obtain the sum 11 (e.g., $4+7$, $5+6$); exactly two ways to obtain the sum 12 (e.g., $4+8$, $5+7$); exactly two ways to obtain the sum 13 (e.g., $5+8$, $6+7$); exactly one way to obtain the sum 14 (e.g., $6+8$); and exactly one way to obtain the sum 15 (e.g., $7+8$).

Equation 7.5.4 can be rewritten, as follows, with the use of Gaussian polynomials as:

$$\begin{aligned}
G_0(x,y,3,8) &= x^0\begin{bmatrix}5\\0\end{bmatrix}_x y^0 + x^4\begin{bmatrix}5\\1\end{bmatrix}_x y^1 + x^9\begin{bmatrix}5\\2\end{bmatrix}_x y^2 + x^{15}\begin{bmatrix}5\\3\end{bmatrix}_x y^3 + x^{22}\begin{bmatrix}5\\4\end{bmatrix}_x y^4 + x^{30}\begin{bmatrix}5\\5\end{bmatrix}_x y^5 \\
&= 1 + x^4\begin{bmatrix}5\\1\end{bmatrix}_x y^1 + x^9\begin{bmatrix}5\\2\end{bmatrix}_x y^2 + x^{15}\begin{bmatrix}5\\3\end{bmatrix}_x y^3 + x^{22}\begin{bmatrix}5\\4\end{bmatrix}_x y^4 + x^{30}\begin{bmatrix}5\\5\end{bmatrix}_x y^5.
\end{aligned}$$

The expanded version of Equation 7.5.2 on page 300, the equation for $G_1(x, z, 3)$, appears below as Equation 7.5.5. It is very similar to the expanded form of the equation for $G_0(x, z, 3, 8)$.

$$\begin{aligned} G_1(x, z, 3) &= \prod_{i=1}^{3}(1 + x^i z) \\ &= (1 + xz)(1 + x^2 z)(1 + x^3 z) \\ &= (1)z^0 + (x^1 + x^2 + x^3)z^1 + (x^3 + x^4 + x^5)z^2 + (x^6)z^3. \end{aligned} \tag{7.5.5}$$

The interpretation of the last line of the equation for $G_1(x, z, 3)$ is discussed in the remainder of this paragraph. The only sum that can be constructed from selecting no elements of the set $\{1, 2, 3\}$ is 0. If only one element can be selected, then the sum must be either 1, 2, or 3. If exactly two elements are selected, without replacement, then the possible sums are 3, 4, and 5. Finally, the only sum that is possible when exactly three elements are selected, without replacement, is 6.

Note that the distribution of sum values that are associated with the coefficients of the various y^i and z^j in $G_0(x, y, 3, 8)$ and $G_1(x, z, 3)$, respectively, appear to be symmetrical, start off being non-monotonically decreasing and, after the midpoint of the distribution is reached, become non-monotonically increasing. This is an instance of unimodaility.

Equation 7.5.5 can be rewritten, as follows, with the aid of Gaussian polynomials as:

$$\begin{aligned} G_1(x, y, 3) &= x^0 \begin{bmatrix} 3 \\ 0 \end{bmatrix}_x z^0 + x^1 \begin{bmatrix} 3 \\ 1 \end{bmatrix}_x z^1 + x^3 \begin{bmatrix} 3 \\ 2 \end{bmatrix}_x z^2 + x^6 \begin{bmatrix} 3 \\ 3 \end{bmatrix}_x z^3 \\ &= 1 + x^1 \begin{bmatrix} 3 \\ 1 \end{bmatrix}_x z^1 + x^3 \begin{bmatrix} 3 \\ 2 \end{bmatrix}_x z^2 + x^6 \begin{bmatrix} 3 \\ 3 \end{bmatrix}_x z^3. \end{aligned} \tag{7.5.6}$$

The convolution of $G_0(x, y, 3, 8)$ and $G_1(x, z, 3)$ yields

$$G(x, y, z, 3, 8) \quad = \quad G_0(x, y, 3, 8) \cdot G_1(x, z, 3)$$

$$
\begin{aligned}
= \quad & (x^0)y^0z^0 + \\
& (x^1 + x^2 + x^3)y^0z^1 + \\
& (x^3 + x^4 + x^5)y^0z^2 + \\
& (x^6)y^0z^3 + \\
& (x^4 + x^5 + x^6 + x^7 + x^8)y^1z^0 + \\
& (x^5 + 2x^6 + 3x^7 + 3x^8 + 3x^9 + 2x^{10} + x^{11})y^1z^1 + \\
& (x^7 + 2x^8 + 3x^9 + 3x^{10} + 3x^{11} + 2x^{12} + x^{13})y^1z^2 + \\
& (x^{10} + x^{11} + x^{12} + x^{13} + x^{14})y^1z^3 + \\
& (x^9 + x^{10} + 2x^{11} + 2x^{12} + 2x^{13} + x^{14} + x^{15})y^2z^0 + \\
& (x^{10} + 2x^{11} + 4x^{12} + 5x^{13} + 6x^{14} + 5x^{15} + 4x^{16} + 2x^{17} + x^18)y^2z^1 + \\
& (x^{12} + 2x^{13} + 4x^{14} + 5x^{15} + 6x^{16} + 5x^{17} + 4x^{18} + 2x^{19} + x^{20})y^2z^2 + \\
& (x^{15} + x^{16} + 2x^{17} + 2x^{18} + 2x^{19} + x^{20} + x^{21})y^2z^3 + \\
& (x^{15} + x^{16} + 2x^{17} + 2x^{18} + 2x^{19} + x^{20} + x^{21})y^3z^0 + \\
& (x^{16} + 2x^{17} + 4x^{18} + 5x^{19} + 6x^{20} + 5x^{21} + 4x^{22} + 2x^{23} + x^{24})y^3z^1 + \\
& (x^{18} + 2x^{19} + 4x^{20} + 5x^{21} + 6x^{22} + 5x^{23} + 4x^{24} + 2x^{25} + x^{26})y^3z^2 + \\
& (x^{21} + x^{22} + 2x^{23} + 2x^{24} + 2x^{25} + x^{26} + x^{27})y^3z^3 + \\
& (x^{22} + x^{23} + x^{24} + x^{25} + x^{26})y^4z^0 + \\
& (x^{23} + 2x^{24} + 3x^{25} + 3x^{26} + 3x^{27} + 2x^{28} + x^{29})y^4z^1 + \\
& (x^{25} + 2x^{26} + 3x^{27} + 3x^{28} + 3x^{29} + 2x^{30} + x^{31})y^4z^2 + \\
& (x^{28} + x^{29} + x^{30} + x^{31} + x^{32})y^4z^3 + \\
& (x^{30})y^5z^0 + \\
& (x^{31} + x^{32} + x^{33})y^5z^1 + \\
& (x^{33} + x^{34} + x^{35})y^5z^2 + \\
& (x^{36})y^5z^3.
\end{aligned}
\tag{7.5.7}
$$

From this equation, we see that the coefficient of the term y^1z^2 (this corresponds to the situation where $r_0 = 1$ and $r_1 = 2$) is

$$T(x) = x^7 + 2x^8 + 3x^9 + 3x^{10} + 3x^{11} + 2x^{12} + x^{13}. \tag{7.5.8}$$

This informs us that the total search lengths range from 7 to 13, inclusive. Additionally, we see that only one ordering had a total search length of 7 and only one had a total search length of 13; that there were two orderings that had total search lengths of 8 and two that had total search lengths of 12; and that there were three orderings each for search lengths of 10, 11, and 12. Notice, also, that encoded in the expansion of $G(x, y, z)$, is distribution information not just for the case where the exponent of $y = r_0$ is 1 and the exponent of $z = r_1$ is 2, but for all the other situations where $r_0 \in \{0, 1, \ldots, n_0\}$ and $r_1 \in \{0, 1, \ldots, n_1\}$.

G$(x, y, z, 3, 8)$ can be rewritten, as follows, with the use of Gaussian polynomials:

$$\begin{aligned}
G(x,y,z,3,8) &= G_0(x,y,3,8)\cdot G_1(x,z,3)\\
&= x^0\begin{bmatrix}5\\0\end{bmatrix}_x y^0\left(x^0\begin{bmatrix}3\\0\end{bmatrix}_x z^0 + x^1\begin{bmatrix}3\\1\end{bmatrix}_x z^1 + x^3\begin{bmatrix}3\\2\end{bmatrix}_x z^2 + x^6\begin{bmatrix}3\\3\end{bmatrix}_x z^3\right) +\\
&\quad x^4\begin{bmatrix}5\\1\end{bmatrix}_x y^1\left(x^0\begin{bmatrix}3\\0\end{bmatrix}_x z^0 + x^1\begin{bmatrix}3\\1\end{bmatrix}_x z^1 + x^3\begin{bmatrix}3\\2\end{bmatrix}_x z^2 + x^6\begin{bmatrix}3\\3\end{bmatrix}_x z^3\right) +\\
&\quad x^9\begin{bmatrix}5\\2\end{bmatrix}_x y^2\left(x^0\begin{bmatrix}3\\0\end{bmatrix}_x z^0 + x^1\begin{bmatrix}3\\1\end{bmatrix}_x z^1 + x^3\begin{bmatrix}3\\2\end{bmatrix}_x z^2 + x^6\begin{bmatrix}3\\3\end{bmatrix}_x z^3\right) +\\
&\quad x^{15}\begin{bmatrix}5\\3\end{bmatrix}_x y^3\left(x^0\begin{bmatrix}3\\0\end{bmatrix}_x z^0 + x^1\begin{bmatrix}3\\1\end{bmatrix}_x z^1 + x^3\begin{bmatrix}3\\2\end{bmatrix}_x z^2 + x^6\begin{bmatrix}3\\3\end{bmatrix}_x z^3\right) +\\
&\quad x^{22}\begin{bmatrix}5\\4\end{bmatrix}_x y^4\left(x^0\begin{bmatrix}3\\0\end{bmatrix}_x z^0 + x^1\begin{bmatrix}3\\1\end{bmatrix}_x z^1 + x^3\begin{bmatrix}3\\2\end{bmatrix}_x z^2 + x^6\begin{bmatrix}3\\3\end{bmatrix}_x z^3\right) +\\
&\quad x^{30}\begin{bmatrix}5\\5\end{bmatrix}_x y^5\left(x^0\begin{bmatrix}3\\0\end{bmatrix}_x z^0 + x^1\begin{bmatrix}3\\1\end{bmatrix}_x z^1 + x^3\begin{bmatrix}3\\2\end{bmatrix}_x z^2 + x^6\begin{bmatrix}3\\3\end{bmatrix}_x z^3\right).
\end{aligned}$$

In this example, our main interest is in the variability of the mean search lengths, rather than the total search lengths, for the orderings. Since each ordering has $r_1 + r_0 = 3$

relevant documents, we need to adapt $T(x)$ to take this into account. The adaptation involves dividing each exponent (which represents the total search length of an ordering) in this function by 3. The resultant ordinary generating function for the discrete variable X is the function

$$M(x) = x^{7/3} + 2x^{8/3} + 3x^{9/3} + 3x^{10/3} + 3x^{11/3} + 2x^{12/3} + x^{13/3}.$$

This function encodes distribution information about the mean search length for the orderings.

If we assume that each of the 15 lengths are equally likely, then $p_M(x)$, the probability generating function for $M(x)$, is

$$\begin{aligned} p_M(x) &= \frac{1}{15}M(x) \\ &= (x^{7/3} + 2x^{8/3} + 3x^{9/3} + 3x^{10/3} + 3x^{11/3} + 2x^{12/3} + x^{13/3})/15. \end{aligned} \tag{7.5.9}$$

From the previous discussion, it is easy to see that

$$\begin{aligned} T(1) &= 1^7 + 2\cdot 1^8 + 3\cdot 1^9 + 3\cdot 1^{10} + 3\cdot 1^{11} + 2\cdot 1^{12} + 1^{13} \\ &= 1+2+3+3+3+2+1 \\ &= 15 \\ &= C(5,1)\cdot C(3,2). \end{aligned}$$

The first and second derivatives of $p_M(x)$, with respect to x, are

$$p'_M(x) = \frac{1}{15}\left(\frac{7x^{4/3}}{3} + \frac{16x^{5/3}}{3} + 9x^2 + 10x^{7/3} + 11x^{8/3} + 8x^3 + \frac{13x^{10/3}}{3}\right)$$

and

$$p''_M(x) = \frac{1}{15}\left(\frac{28x^{1/3}}{9} + \frac{80x^{2/3}}{9} + 18x + \frac{70x^{4/3}}{3} + \frac{88x^{5/3}}{3} + 24x^2 + \frac{130x^{7/3}}{9}\right),$$

respectively.

The first derivative, when evaluated at $x = 1$, computes μ, the mean.Therefore,

$$\begin{aligned} \mu &= p'_M(1) \\ &= (7/3 + 16/3 + 9 + 10 + 11 + 8 + 13/3)/15 \\ &= 50/15 \\ &= 10/3. \end{aligned}$$

The second derivative, when evaluated at $x = 1$, is

$$\begin{aligned} p''_M(1) &= (28/9 + 80/9 + 18 + 70/3 + 88/3 + 24 + 130/9)/15 \\ &= (1090/9)/15 \\ &= 218/27. \end{aligned}$$

The population variance, σ^2, can be computed as follows:

$$\begin{aligned} \sigma^2 &= p''_M(1) + p'_M(1) - p'_M(1)^2 \\ &= 218/27 + 10/3 - (10/3)^2 \\ &= 218/27 + 10/3 - 100/9 \\ &= 8/27. \end{aligned}$$

Hence, the population standard deviation is $\sigma = \sqrt{8/27}$. Since a sample variance s^2, for a population of size N, always differs from the corresponding population variance by a factor of $N/(N-1)$, the sample variance and sample standard deviation s are $s^2 = 8/27 \cdot 15/14 = 20/63$ and $s = \sqrt{20/63}$, respectively.

Since the entire population is known, we could simply use the population variance as our variance. However, if we are using the variance for inferential, as contrasted to descriptive, purposes then we may want to be a little more conservative, and, instead, use the sample variance to help construct the confidence intervals for the total search lengths and mean search lengths.

The coefficients of the various $y^{r_0}z^{r_1}$ in the expansion of $G(x, y, z)$, on page 304, seem to indicate that the distribution of the TSL values for each coefficient are *palindromic*, that is, the sequence of values for the x-values read the same from left to right as they do from right to left. For example, in the expression that corresponds to z^1y^1, the successive frequency counts are 1, 2, 3, 3, 3, 2, 1 for the TSLs 5, 6, 7, 8, 9, 10, 11, respectively. This is an example of reciprocity. Furthermore, not only do we see symmetry of the frequency counts around the midpoints of the distinct TSL values, when they are arranged, in order, from the minimum to the maximum, but, we also notice that the frequency counts are monotonically non-decreasing from the minimum TSL value to the midpoint TSL value and that the frequency counts are monotonically non-increasing from the midpoint value to the maximum TSL value. This is an example of unimodality.

If we can prove that the symmetry and monotonicity attributes always hold, then this is invaluable to us in our validation efforts because that means that it is appropriate to use a parametric test such as the *t-distribution* (Walpole, 2002), the *normal distribution* (Walpole, 2002), or the beta distribution (Pratt et al., 1995), depending on the size of the population and other considerations.

7.5.2 Two Functions That Calculate the Sums of the Minimum and Maximum k Values in a Range of Integers

Definition 7.5.2.1. The *sum-of-the-minimum-k-values* function

$$\mathrm{minSum} : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

with parameters $n, k, s \in \mathbb{N}$ in positions 1, 2, and 3, respectively, in the parameter list, and $n \geq k$, yields a value that is equal to the sum of the k consecutive nonnegative integers that are in the range $[1 + s, k + s]$. This value is equal to $k(1 + k)/2 + sk$.

Lemma 7.5.1. *The sum of the k consecutive nonnegative integers that are in the range $[1 + s, k + s]$ is equal to $k(1 + k)/2 + sk$.*

Proof.

$$\begin{aligned}\sum_{i=1+s}^{k+s} i &= \sum_{i=1}^{k+s} i - \sum_{i=1}^{s} i \\ &= (k + s)(k + s + 1)/2 - s(s + 1)/2 \\ &= k(1 + 2s + k)/2 \\ &= k(1 + k)/2 + sk.\end{aligned}$$

□

Definition 7.5.2.2. The *sum-of-the-maximum-k-values* function

$$\mathrm{maxSum} : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

with parameters $n, k, s \in \mathbb{N}$ in positions 1, 2, and 3, respectively, in the parameter list, and $n \geq k$, yields a value that is equal to the sum of the k consecutive nonnegative integers that are in the range $[n - k + 1 + s, n + s]$. This value is equal to $k(1 - k)/2 + (n + s)k$.

Lemma 7.5.2. *The sum of the k consecutive nonnegative integers that are in the range $[n-k+1+s, n+s]$ is equal to $k(1+k)/2+sk$.*

Proof.

$$\begin{aligned}\sum_{i=n-k+1+s}^{n+s} i &= \sum_{i=1}^{n+s} i - \sum_{i=1}^{n-k+s} i \\ &= (n+s)(n+s+1)/2 - (n-k+s)(n-k+s+1)/2 \\ &= k(1-k+2n+2s)/2 \\ &= k(1-k)/2 + (n+s)k.\end{aligned}$$

□

Definition 7.5.2.3. The *difference-of-sums-of-the-minimum-and-maximum-k-values* function

$$\text{diffSum} : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

with parameters $n, k, s \in \mathbb{N}$ in positions 1, 2, and 3, respectively, in the parameter list, and $n \geq k$, yields a value that is equal to the difference of the sums of the k consecutive nonnegative integers that are in the ranges $[1+s, k+s]$ and $[n-k+1+s, n+s]$. This value is equal to $k(n-k)$.

Lemma 7.5.3. *The difference of the sum of the k consecutive integers in the range $[1+s, k+s]$ and the k consecutive ones in the range $[n-k+1+s, n+s]$ is equal to $k(n-k)$.*

Proof.

$$\begin{aligned}\text{diffSum}(n,k,s) &= \text{maxSum}(n,k,s) - \text{minSum}(n,k,s) \\ &= k(1-k)/2 + (n+s)k - (k(1+k)/2 + sk)\end{aligned}$$

$$
\begin{aligned}
&= k(1-k)/2 - k(1+k)/2 + (n+s)k - sk \\
&= k(1-k-1-k)/2 + (n+s)k - sk \\
&= -k^2 + nk + sk - sk \\
&= k(n-k).
\end{aligned}
$$

□

7.5.3 The Example Continued — The Distribution of Total Search Length Values For Feature Frequency 0

The minimum TSL value for documents with feature frequency 0, minTSL_0, corresponds to the situation where the relevant documents occupy positions $n_1+1, n_1+2, \ldots, n_1+r_0$, inclusive, in an ordering. The maximum TSL value, maxTSL_0, corresponds to the the situation where the relevant documents occupy positions $N-r_0+1, N-r_0+2, \ldots, N$, inclusive, in an ordering.

The TSL values for y^1, when $n_0 = 5$, $n_1 = 3$, $r_0 = 1$, and $r_1 = 2$, are in the closed interval $[4, 8]$. This is evidenced by the calculations below. The minimum TSL value for those documents that do not contain the query term is

$$
\begin{aligned}
\text{minSum}(n_0, r_0, n_1) &= \text{minSum}(5, 1, 3) \\
&= 1(1+1)/2 + 3 \cdot 1 \\
&= 4
\end{aligned}
$$

and the maximum one is

$$
\begin{aligned}
\text{maxSum}(n_0, r_0, n_1) &= \text{maxSum}(5, 1, 3) \\
&= 1(1-1)/2 + (5+3) \cdot 1
\end{aligned}
$$

$$= 8.$$

It is also evidenced by the expression $(x^4 + x^5 + x^6 + x^7 + x^8)y^1$ in the original version of $G_0(x, y, 3, 8)$ and by the extended version of the Cauchy binomial theorem. By this theorem, the coefficient in $G_0(x, y, 3, 8)$ that corresponds to the situation where only one element of the set $\{4, 5, 6, 7, 8\}$ can be chosen is

$$\begin{aligned} q^{\binom{r_0+1}{2}+r_0\cdot n_1}\begin{bmatrix} n_0 \\ r_0 \end{bmatrix}_q &= q^{\binom{1+1}{2}+1\cdot 3}\begin{bmatrix} 5 \\ 1 \end{bmatrix}_q \\ &= q^4(1 + q + q^2 + q^3 + q^4) \\ &= q^4 + q^5 + q^6 + q^7 + q^8. \end{aligned}$$

7.5.4 The Example Continued — The Distribution of Total Search Length Values For Feature Frequency 1

The minimum TSL value for documents with feature frequency 1, minTSL_1, corresponds to the the situation where the relevant documents occupy positions $1, 2, \ldots, r_0$, inclusive, in an ordering. The maximum TSL value, maxTSL_1, corresponds to the situation where the relevant documents occupy positions $n_1 - r_1 + 1, n_1 - r_1 + 2, \ldots, n_1$, inclusive, in an ordering.

By the calculations below, the TSL values for z^2, when $n_0 = 5$, $n_1 = 3$, $r_0 = 1$, and $r_1 = 2$, are in the closed interval $[3, 5]$. The minimum TSL value for the documents that contain the query term is

$$\begin{aligned} \text{minSum}(n_1, r_1, 0) &= \text{minSum}(3, 2, 0) \\ &= 2(1 + 2)/2 + 0 \cdot 2 \\ &= 3, \end{aligned}$$

and the maximum one is

$$\begin{aligned}\mathrm{maxSum}(n_1, r_1, 0) &= \mathrm{maxSum}(3, 2, 0)\\ &= 2(1-2)/2 + (3+0)\cdot 2\\ &= 5.\end{aligned}$$

It is also evidenced by the expression $(x^3+x^4+x^5)z^2$, in the original version of $G_1(x, y, 3)$, and by the extended version of the Cauchy binomial theorem. By this theorem, the coefficient in $G_1(x, y, 3)$, that corresponds to the situation where exactly two distinct elements of the set $\{1, 2, 3\}$ can be chosen, is

$$\begin{aligned}q^{\binom{r_1+1}{2}}\begin{bmatrix}n_1\\ r_1\end{bmatrix}_q &= q^{\binom{2+1}{2}}\begin{bmatrix}3\\ 2\end{bmatrix}_q\\ &= q^3(1+q+q^2)\\ &= q^3+q^4+q^5.\end{aligned}$$

7.5.5 The Example Continued — The Combined Distribution of Total Search Length Values

The combined distribution is described by the expression

$$\begin{aligned}q^{\binom{r_0+1}{2}+r_0\cdot n_1+\binom{r_1+1}{2}}\begin{bmatrix}n_0\\ r_0\end{bmatrix}_q\begin{bmatrix}n_1\\ r_1\end{bmatrix} &= q^{\binom{r_0+1}{2}+r_0\cdot n_1+\binom{r_1+1}{2}}\begin{bmatrix}n_0\\ r_0\end{bmatrix}_q\begin{bmatrix}n_1\\ r_1\end{bmatrix}_q\\ &= q^{\binom{1+1}{2}+1\cdot 3+\binom{2+1}{2}}\begin{bmatrix}5\\ 1\end{bmatrix}_q\begin{bmatrix}3\\ 2\end{bmatrix}_q\\ &= q^{1+3+3}\begin{bmatrix}5\\ 1\end{bmatrix}_q\begin{bmatrix}3\\ 2\end{bmatrix}_q\\ &= q^7(1+q+q^2+q^3+q^4)(1+q+q^2)\\ &= q^7(1+2q+3q^2+3q^3+3q^4+2q^5+q^6)\end{aligned}$$

$$= q^7 + 2q^8 + 3q^9 + 3q^{10} + 3q^{11} + 2q^{12} + q^{13}.$$

This corresponds with the expression $(x^7 + 2x^8 + 3x^9 + 3x^{10} + 3x^{11} + 2x^{12} + x^{13})y^1z^2$ from Equation 7.5.7 on page 304. The expression

$$x^7 + 2x^8 + 3x^9 + 3x^{10} + 3x^{11} + 2x^{12} + x^{13}$$

is identical to the expression that defines $T(x)$, the total search length, in Equation 7.5.8 on page 305. The interpretation of this is that the 8 document collection, with parameters $N = n_0 + n_1$, where $n_0 = r_0 + s_0$, $n_1 = r_1 + s_1$, $r_0 = 1$, $s_0 = 4$, $r_1 = 2$, and $s_1 = 1$, over all the possible sequences of documents, has 7 distinct total search lengths over the

$$1 + 2 + 3 + 3 + 3 + 2 + 1 = 15$$

possible sequences. Note that, just like with Equation 7.5.7 on page 304, this expression informs us that the seven distinct total search lengths range from 7 to 13, inclusive. Also, we see that only one ordering had a total search length of 7 and only one had a total search length of 13; that there were two orderings that had total search lengths of 8 and two that had total search lengths of 12; and that there were three orderings each for search lengths of 10, 11, and 12.

7.6 Useful Definitions and Theorems

These theorems cover several aspects of expected values, variances, covariances, and some of their linear transformations. They are useful for the discussions and formula development in the remainder of this chapter. The proofs of these theorems are provided in the source(s) cited for each theorem at its end. The notations and styles of exposition used in the sources differed somewhat from each other and, because of that, the author of this dissertation developed a consistent notation and style that is used to re-express these theorems.

Definition 7.6.0.1. If X is a discrete random variable with probability distribution $f(x)$, then the *mean* or *expected value* of X is

$$\mu = E[X] = \sum_x xf(x)$$

Walpole (2002).

Definition 7.6.0.2. If X is a discrete random variable with probability distribution $f(x)$ and mean μ, then the *variance* of X is

$$\sigma^2 = E[(X-\mu)^2] = \sum_x (x-\mu)^2 f(x).$$

The positive square root of the variance, σ, is called the *standard deviation* of X

Walpole (2002).

Theorem 7.6.1. *If X is a discrete random variable with mean μ, then the* variance *of X can also be expressed as*

$$\sigma^2 = E[X^2] - \mu^2$$

Walpole (2002).

Definition 7.6.0.3. If X and Y are random variables, then the *covariance* of X and Y is

$$\text{Cov}[X,Y] = E[\text{XY}] - E[X]E[Y]$$

Walpole (2002).

Theorem 7.6.2. *If X and Y are two independent random variables, then*

$$\mathrm{Cov}[X, Y] = 0$$

Walpole (2002).

Theorem 7.6.3. *If a and b are constants, X is a random variable, and $E[X]$ is the expected value of X , then*

$$E[aX + b] = aE[X] + b$$

Walpole (2002).

Theorem 7.6.4. *If a and b are constants, X is a random variable, and* $\mathrm{Var}[X]$ *is the* variance *of X, then*

$$\mathrm{Var}[aX + b] = a^2\mathrm{Var}[X]$$

Walpole (2002).

Theorem 7.6.5. *If X and Y are are random variables, and $g(X, Y)$ and $h(X, Y)$ are functions of these variables, then*

$$E[g(X, Y) \pm h(X, Y)] = E[g(X, Y)] \pm E[h(X, Y)]$$

Walpole (2002).

Theorem 7.6.6. *If $X_1, X_2, \ldots, X_n$ are independent random variables, and $a_1, a_2, \ldots, a_n$ are constants, then*

$$Var[a_1x_1 + a_2x_2 + \cdots + a_nx_n] = a_1^2Var[X_1] + a_2^2Var[X_2] + \cdots + a_n^2Var[X_n]$$

Walpole (2002).

Theorem 7.6.7. *If X and Y are two independent random variables, then*

$$E[XY] = E[X]E[Y]$$

Walpole (2002).

Theorem 7.6.8. *If X and Y are random variables for which* $\mathrm{Var}[\mathrm{XY}]$ *exists, then*

$$E[\mathrm{XY}] = E[X]E[Y] + \mathrm{Cov}[X,Y]$$

and

$$\begin{aligned}\mathrm{Var}[\mathrm{XY}] = {} & (E[Y])^2\mathrm{Var}[X] \\ & + (E[X])^2\mathrm{Var}[Y] \\ & + 2E[X]E[Y]\mathrm{Cov}[X,Y] \\ & - (\mathrm{Cov}[X,Y])^2 \\ & + E[(X-E[X])^2(Y-E[Y])^2] \\ & + 2E[Y]E[(X-E[X])^2(Y-E[Y])] \\ & + 2E[X]E[(X-E[X])(Y-E[Y])^2]\end{aligned}$$

Mood et al. (1973); Blumenfeld (2001).

7.7 Expected Value and Variance of the Normalized Search Length

The main result of this section is a proof that the value of $\mathcal{A}$ is equal to the expected value of the normalized search length. Another important result is an expression that can be used to calculate the variance that is associated with the normalized search length.

The probability mass functions (*pmf*s) for the total search lengths are

$$\begin{aligned} p_{\mathrm{T},0}(x) &= \frac{[y^{r_0}]G_0(x, y, n_1, N)}{\binom{n_0}{r_0}}, \\ p_{\mathrm{T},1}(x) &= \frac{[z^{r_1}]G_1(x, z, n_1)}{\binom{n_1}{r_1}}, \text{ and} \\ p_{\mathrm{T,G}}(x) &= \frac{[y^{r_0}z^{r_1}]\mathrm{G}(x, y, z, n_1, N)}{\binom{n_0}{r_0}\binom{n_1}{r_1}}. \end{aligned}$$

These are polynomials that have degrees $\mathrm{maxSum}(n_0, r_0, n_1)$, $\mathrm{maxSum}(n_1, r_1, 0)$, and $\mathrm{maxSum}(n_0, r_0, n_1) + \mathrm{maxSum}(n_1, r_1, 0)$, respectively. The respective means are $p'_{\mathrm{T},0}(1)$, $p'_{\mathrm{T},1}(1)$, and $p'_{\mathrm{T,G}}(1)$. Similarly, the respective variances are

$$\begin{aligned} &p''_{\mathrm{T},0}(1) + p'_{\mathrm{T},0}(1) - p'_{\mathrm{T},0}(1)^2, \\ &p''_{\mathrm{T},1}(1) + p'_{\mathrm{T},1}(1) - p'_{\mathrm{T},1}(1)^2, \text{ and} \\ &p''_{\mathrm{T,G}}(1) + p'_{\mathrm{T,G}}(1) - p'_{\mathrm{T,G}}(1)^2. \end{aligned}$$

The corresponding means and variances for the mean search lengths can be obtained in one of two ways: (1) alter the exponents of the addends in the pmfs for the TSLs to reflect that the pmf is for an MSL rather than a TSL or (2) calculate the means and variances for the TSLs, but scale them afterwards to obtain the means and variances for the MSLs.

The first way transforms a TSL pmf into an MSL pmf by dividing the exponent of each of the TSL's addends by the number of relevant documents appropriate for that pmf. These numbers are r_0, r_1, and $r_0 + r_1$, respectively, for the TSLs that correspond to the relevant documents that do not contain the query term, to the relevant documents that contain the query term, and to all of the relevant documents.

Therefore, the pmfs for the MSLs are

$$\begin{aligned}
p_{\mathrm{M},0}(x) &= \frac{\operatorname{divexp}\left([y^{r_0}]G_0(x,y,n_1,N), r_0\right)}{\binom{n_0}{r_0}}, \\
p_{\mathrm{M},1}(x) &= \frac{\operatorname{divexp}\left([z^{r_1}]G_1(x,z,n_1), r_1\right)}{\binom{n_0}{r_1}}, \text{ and} \\
p_{\mathrm{M},\mathrm{G}}(x) &= \frac{\operatorname{divexp}\left([y^{r_0}z^{r_1}]\mathrm{G}(x,y,z,n_1,N), r_0+r_1\right)}{\binom{n_0}{r_0}\binom{n_1}{r_1}}
\end{aligned}$$

where each pmf is a function $f(x)$ such that $f(x) = \sum_{i=1}^{b-a+1} c_i x^{e_i}$, where the smallest and largest exponents of x in $f(x)$ are a and b, respectively; the coefficient of the ith addend is c_i; the exponent of the ith addend is $e_i = a + 1 - i$; and $\operatorname{divexp}(f(x), d) = \sum_{i=1}^{b-a+1} c_i x^{e_i/d}$ is the result of dividing each exponent of $f(x)$ by d.

The second way to calculate MSLs from TSLs makes use of two well-known statistical transformations (Walpole, 2002) on random variables: one for the mean (Equation 7.6.3 on page 316) and the the other for the variance (Equation 7.6.4 on page 316).

Lemma 7.7.1. *The means that are associated with $p_{\mathrm{M},0}(x)$, $p_{\mathrm{M},1}(x)$, and $p_{\mathrm{M},\mathrm{G}}(x)$ are the same as those that are associated with*

$$p_{\mathrm{T},0}(x)/r_0,\ \ p_{\mathrm{T},1}(x)/r_1,\ \ \textit{and}\ p_{\mathrm{T},\mathrm{G}}(x)/(r_0+r_1),$$

respectively.

Proof. Let $b = 0$, $a = 1/r_0$, and $X = \mathrm{TSL}_0$. Then, by Identity 7.6.3 on page 316, the mean that is associated with $p_{\mathrm{M},0}(x)$ is the same as the one one that is associated with $p_{\mathrm{T},0}(x)/r_0$. The proofs for $p_{\mathrm{M},1}(x)$ and $p_{\mathrm{M},\mathrm{G}}(x)$ are similar and are not discussed here. □

Lemma 7.7.2. *The variances that are associated with $p_{\mathrm{M},0}(x)$, $p_{\mathrm{M},1}(x)$, and $p_{\mathrm{M},\mathrm{G}}(x)$ are* *the same as those that are associated with*

$$p_{\mathrm{T},0}(x)/r_0^2,\ p_{\mathrm{T},1}(x)/r_1^2,\ \textit{and}\ p_{\mathrm{T,G}}(x)/(r_0+r_1)^2,$$

respectively.

Proof. Let $b = 0$, $a = 1/r_0$, and $X = \mathrm{TSL}_0$. Then, by Theorem 7.6.4 on page 316, the variance that is associated with $p_{\mathrm{M},0}(x)$ is the same as the variance that is associated with $p_{\mathrm{T},0}(x)/r_0^2$. The proofs for $p_{\mathrm{M},1}(x)$ and $p_{\mathrm{M,G}}(x)$ are similar and are not discussed here. □

The MSL values are unnormalized. Let $\mathcal{M}_i$ denote the unnormalized MSL value for an individual ordering $i \in O$, where O is the set of all orderings for a collection of

$$N = r_0 + r_1 + s_0 + s_1$$

documents, and let $\ddot{\mathcal{M}}_i$ denote the corresponding normalized value. Any unnormalized MSL value can be normalized by subtracting 1/2 from it and, then, dividing that result by the number of documents in the collection (this transformation makes use of the results from Equation 7.2.9 on page 282 and Equation 7.2.10 on page 283). Therefore, $\ddot{\mathcal{M}}_i$, the normalized version of the $\mathcal{M}_i$ value for an individual ordering i can be computed by this transformation:

$$\ddot{\mathcal{M}}_i = (\mathcal{M}_i - 1/2)/N.$$

Lemma 7.7.3. *The expected value, $E[\ddot{\mathcal{M}}]$, of the random variable $\ddot{\mathcal{M}}$ is $\mathcal{A}$. Its range is in the closed interval $[0, 1]$.*

Proof. Let $R = r_0 + r_1$, let $c = C(n_0, r_0) \times C(n_1, r_1)$, and let O be the set of possible orderings for an N document collection with $N = r_0 + r_1 + s_0 + s_1$. Then

$$\begin{aligned}
E[\ddot{\mathcal{M}}] &= \Sigma_{i \in O}[((\mathcal{M}_i - 1/2)/N) \cdot \Pr(i \in O)] \\
&= \Sigma_{i \in O}\left[(\mathcal{M}_i - 1/2)/N) \cdot \frac{1}{c}\right] \\
&= (\Sigma_{i \in O}(\mathcal{M}_i - 1/2))/(cN) \\
&= (cN)^{-1}(\Sigma_{i \in O}\mathcal{M}_i - \Sigma_{i \in O}1/2) \\
&= (cN)^{-1}(\Sigma_{i \in O}\mathcal{M}_i - c/2) \\
&= ((cN)^{-1}\Sigma_{i \in O}\mathcal{M}_i) - \frac{1}{2N} \\
&= ((cN)^{-1}S_{\text{rel}}/R) - \frac{1}{2N} \\
&= (cNR)^{-1}S_{\text{rel}} - \frac{1}{2N} \\
&= (cNR)^{-1}\left[\frac{1}{2}\binom{n_1}{r_1}\binom{n_0}{r_0}[(r_1 + r_0)(n_1 + 1)] + \frac{1}{2}\binom{n_1}{r_1}\binom{n_0}{r_0}r_0 N\right] - \frac{1}{2N} \\
&= (cNR)^{-1}\left[\frac{1}{2}c\,[R(n_1 + 1)] + \frac{1}{2}c \cdot r_0 N\right] - \frac{1}{2N} \\
&= (NR)^{-1}\left[\frac{1}{2}\,[R(n_1 + 1)] + \frac{1}{2}r_0 N\right] - \frac{1}{2N} \\
&= (2NR)^{-1}\,[R(n_1 + 1) + r_0 N] - \frac{R}{2NR} \\
&= (2NR)^{-1}(Rn_1 + R + r_0 N - R) \\
&= \frac{n_1 R + r_0 N}{2NR}.
\end{aligned}$$

Note that the expression in the last line of the derivation is the same as the expression for $\mathcal{A}$ in Equation 7.2.17 on page 284. Therefore,

$$\mathcal{A} = E[\ddot{\mathcal{M}}].$$

$\square$

Lemma 7.7.4. *The variance,* $\mathrm{Var}[\ddot{\mathcal{M}}]$*, of the random variable* $\ddot{\mathcal{M}}$ *is*

$$E[\ddot{\mathcal{M}}^2] - \mathcal{A}^2.$$

Proof.

$$\begin{aligned}\mathrm{Var}[\ddot{\mathcal{M}}] &= E[\ddot{\mathcal{M}}^2] - (E[\ddot{\mathcal{M}}])^2 \\ &= E[\ddot{\mathcal{M}}^2] - \mathcal{A}^2.\end{aligned}$$

□

7.8 Expected Value and Variance of the Unnormalized Search Length

The main result of this section is a proof that the value of $\mathcal{Q}$ for a specific ranking method is equal to the expected value of the unnormalized search length for this method. Another important result is an expression that can be used to calculate the variance that is associated with the unnormalized search length.

Lemma 7.8.1. *The expected value of the random variable*

$$\mathcal{L} = N(\mathcal{B} \cdot \ddot{\mathcal{M}} + (1 - \mathcal{B})(1 - \ddot{\mathcal{M}})) + 1/2$$

is

$$\mathrm{E}[\mathcal{L}] = N(\mathcal{Q}\mathcal{A} + (1 - \mathcal{Q})(1 - \mathcal{A})) + 1/2,$$

where $\mathcal{B}$ *and* $\ddot{\mathcal{M}}$ *are random variables that are assumed to be independent.*

Proof. The expressions $\mathrm{Cov}(\mathcal{B}, \ddot{\mathcal{M}})$ and $\mathrm{Cov}(1 - \mathcal{B}, 1 - \ddot{\mathcal{M}})$ are equal to 0 because of the independence assumption and Theorem 7.6.2 on page 316.

$$\begin{aligned}
E[\mathcal{B} \cdot \ddot{\mathcal{M}}] &= E[\mathcal{B}]E[\ddot{\mathcal{M}}] + \mathrm{Cov}[\mathcal{B}, \ddot{\mathcal{M}}] \\
&= \mathcal{Q}\mathcal{A} + \mathrm{Cov}[\mathcal{B}, \ddot{\mathcal{M}}] \\
&= \mathcal{Q}\mathcal{A} + 0 \\
&= \mathcal{Q}\mathcal{A}. \\
E[(1-\mathcal{B})(1-\ddot{\mathcal{M}})] &= E[1-\mathcal{B}]E[1-\ddot{\mathcal{M}}] \\
&\quad\quad + \mathrm{Cov}[1-\mathcal{B}, 1-\ddot{\mathcal{M}}] \\
&= (1-\mathcal{Q})(1-\mathcal{A}) + \mathrm{Cov}[1-\mathcal{B}, 1-\ddot{\mathcal{M}}] \\
&= (1-\mathcal{Q})(1-\mathcal{A}) + 0 \\
&= (1-\mathcal{Q})(1-\mathcal{A}).
\end{aligned}$$

Therefore,

$$\mathrm{ASL} = E[\mathcal{L}] = N(\mathcal{Q}\mathcal{A} + (1-\mathcal{Q})(1-\mathcal{A})) + 1/2.$$

The justifications for the final equation are Theorem 7.6.3 on page 316, Theorem 7.6.5 on page 316, and Theorem 7.6.7 on page 317. □

Lemma 7.8.2. *The variance of* $\mathcal{B} \cdot \ddot{\mathcal{M}}$*, the product of the random variables* $\mathcal{B}$ *and* $\ddot{\mathcal{M}}$*, assuming statistical independence, is*

$$\mathrm{Var}[\mathcal{B}\ddot{\mathcal{M}}] = \mathcal{A}^2\mathcal{Q}(1-\mathcal{Q}) + \mathcal{Q}\mathrm{Var}[\ddot{\mathcal{M}}].$$

Proof. The variance can be written initially as

$$\begin{aligned}
\mathrm{Var}[\mathcal{B}\ddot{\mathcal{M}}] \quad = \quad & (E[\ddot{\mathcal{M}}])^2 Var[\mathcal{B}] \\
& + (E[\mathcal{B}])^2 Var[\ddot{\mathcal{M}}] \\
& + \mathrm{Var}[\mathcal{B}]Var[\ddot{\mathcal{M}}].
\end{aligned}$$

Substitutions yield

$$\begin{aligned}\mathrm{Var}[\mathcal{B}\ddot{\mathcal{M}}] &= \mathcal{A}^2\mathcal{Q}(1-\mathcal{Q}) \\ &\quad +\mathcal{Q}^2 Var[\ddot{\mathcal{M}}] \\ &\quad +\mathcal{Q}(1-\mathcal{Q})\mathrm{Var}[\ddot{\mathcal{M}}].\end{aligned}$$

Simplifications yield

$$\mathrm{Var}[\mathcal{B}\ddot{\mathcal{M}}] = \mathcal{A}^2\mathcal{Q}(1-\mathcal{Q}) + \mathcal{Q}\mathrm{Var}[\ddot{\mathcal{M}}].$$

□

Lemma 7.8.3. *The variance of* $(1-\mathcal{B})(1-\ddot{\mathcal{M}})$*, the product of linear transformations of random variables* $\mathcal{B}$ *and* $\ddot{\mathcal{M}}$*, assuming statistical independence, is*

$$\mathrm{Var}[(1-\mathcal{B})(1-\ddot{\mathcal{M}})] = (1-\mathcal{A})^2\mathcal{Q}(1-\mathcal{Q}) + (1-\mathcal{Q})\mathrm{Var}[\ddot{\mathcal{M}}].$$

Proof. Assume that $\mathcal{B}$ and $\ddot{\mathcal{M}}$ are independent random variables. This means that $\mathrm{Cov}(\mathcal{B},\ddot{\mathcal{M}}) = 0$ and that $\mathrm{Var}[\mathcal{B}\ddot{\mathcal{M}}]$ can be written as it appears below. The variance can be written initially as

$$\begin{aligned}\mathrm{Var}[(1-\mathcal{B})(1-\ddot{\mathcal{M}})] &= (E[1-\ddot{\mathcal{M}}])^2\mathrm{Var}[1-\mathcal{B}] \\ &\quad +(E[1-\mathcal{B}])^2\mathrm{Var}[1-\ddot{\mathcal{M}}] \\ &\quad +\mathrm{Var}[1-\mathcal{B}]\mathrm{Var}[1-\ddot{\mathcal{M}}].\end{aligned}$$

After applying Theorem 7.6.4 on page 316, this equation results:

$$\mathrm{Var}[(1-\mathcal{B})(1-\ddot{\mathcal{M}})] = (E[1-\ddot{\mathcal{M}}])^2\mathrm{Var}[\mathcal{B}]$$

$$
\begin{aligned}
&+(E[1-\mathcal{B}])^2\mathrm{Var}[\ddot{\mathcal{M}}]\\
&+\mathrm{Var}[\mathcal{B}]\mathrm{Var}[\ddot{\mathcal{M}}].
\end{aligned}
$$

Substitutions yield

$$
\begin{aligned}
\mathrm{Var}[(1-\mathcal{B})(1-\ddot{\mathcal{M}})] \;\; = \;\; &(1-\mathcal{A})^2\mathcal{Q}(1-\mathcal{Q})\\
&+(1-\mathcal{Q})^2\mathrm{Var}[\ddot{\mathcal{M}}]\\
&+\mathcal{Q}(1-\mathcal{Q})\mathrm{Var}[\ddot{\mathcal{M}}].
\end{aligned}
$$

Simplifications yield

$$
\begin{aligned}
\mathrm{Var}[(1-\mathcal{B})(1-\ddot{\mathcal{M}})] \;\; = \;\; &(1-\mathcal{A})^2\mathcal{Q}(1-\mathcal{Q})\\
&+(1-\mathcal{Q})\mathrm{Var}[\ddot{\mathcal{M}}].
\end{aligned}
$$

□

Lemma 7.8.4. *The variance of the random variable*

$$\mathcal{L} = N(\mathcal{B}\cdot\ddot{\mathcal{M}} + (1-\mathcal{B})(1-\ddot{\mathcal{M}})) + 1/2$$

is

$$\mathrm{Var}[\mathcal{L}] = N^2\left(\left(2\mathcal{A}^2 - 2\mathcal{A} + 1\right)\mathcal{Q}(1-\mathcal{Q}) + \mathrm{Var}[\ddot{\mathcal{M}}]\right).$$

Proof. The sum of $\mathrm{Var}[\mathcal{B}\cdot\ddot{\mathcal{M}}]$ and $\mathrm{Var}[(1-\mathcal{B})(1-\ddot{\mathcal{M}})]$ is

$$\left(\mathcal{A}^2 + 1 - 2\mathcal{A} + \mathcal{A}^2\right)\mathcal{Q}(1-\mathcal{Q}) + \mathrm{Var}[\ddot{\mathcal{M}}].$$

due to Theorem 7.6.6 on page 316. After simplification, and the application of Theorem 7.6.4 on page 316, the resultant formula is

$$N^2 \left(\left(2\mathcal{A}^2 - 2\mathcal{A} + 1\right) \mathcal{Q}(1 - \mathcal{Q}) + \mathrm{Var}[\ddot{\mathcal{M}}] \right).$$

□

7.9 Retrieval Status Value, Weights, and Document Ranking

Before a collection of documents can be ranked, in conjunction with a query q; ranking method rm; and parameters r_1, r_0, s_1, s_0, and N; all the documents in the collection must be assigned a *retrieval status value* (RSV). The RSV is a weight of how relevant a document is to a query. The higher this weight, the more relevant a document is estimated to be; the lower the weight, the less relevant a document is estimated to be. When documents are non-ascendingly ordered by the RSV, the most relevant documents are expected to be at the front of the sequence of ranked documents, the least relevant documents are expected to be at the rear.

In the query-document model used in this dissertation, the RSV is the product of the *query term weight* (qtw) and the *document term weight* (dtw), that is,

$$\mathrm{RSV} = \mathrm{qtw} * \mathrm{dtw}.$$

In this model, a document is either relevant or not relevant (binary relevance) and either has a desired feature or does not have it (the dtw of a document, where the feature occurs multiple times, is the same as the dtw of a document where the feature occurs exactly once). For query q and ranking method rm, the dtw is always the same for each document in the collection. According to the information in Table 7.1 on page 329, the value of this weight is always positive for the coordination level matching (CLM) ranking method. Its value for the other 5 methods may be negative, zero, or positive, depending on the values of p, t, and possibly q for the weak 4-composition (r_1, s_0, r_0, s_1). Table 7.9 on page 329 details the relationships between the RSVs and the query and document term weights.

This means that the RSVs for a ranked collection can have, at most, two distinct values. These are the 5 ranking possibilities: RSVs are either (1) all zeros, (2) all positive numbers, (3) all negative numbers, (4) a mixture of zero and positive numbers, or (5) a mixture of zero and negative numbers. If there is a mixture of numbers, there are always two distinct numbers, one of which is always 0. The ranking algorithm divides these documents into two clusters — one cluster solely contains documents that have a value of 0 for their RSV, the other cluster solely contains the documents that do not have a value of 0 for their RSV. These two general ranking orders are depicted in Figure 7.8 on the following page.

Possibility (1) can be viewed as a special case of either of the diagrams in Figure 7.8 when $n_1 = 0$ is true. Possibility (2) is a special case of Figure 7.8(a) when $n_0 = 0$ is true. Similarly, Possibility (3) is a special case of Figure 7.8(b) when $n_0 = 0$ is true. Possibilities (4) and (5) correspond to Figures 7.8(a) and 7.8(b), respectively.

7.10 A Family of ASL Measures

Definition 7.10.0.4. Let $f_{\text{ASL}}(N, q, a) = N(qa+(1-q)(1-a))+1/2$, where N represents the number of documents in a collection, q represents a quality of ranking value, and a represents a normalized search length value.

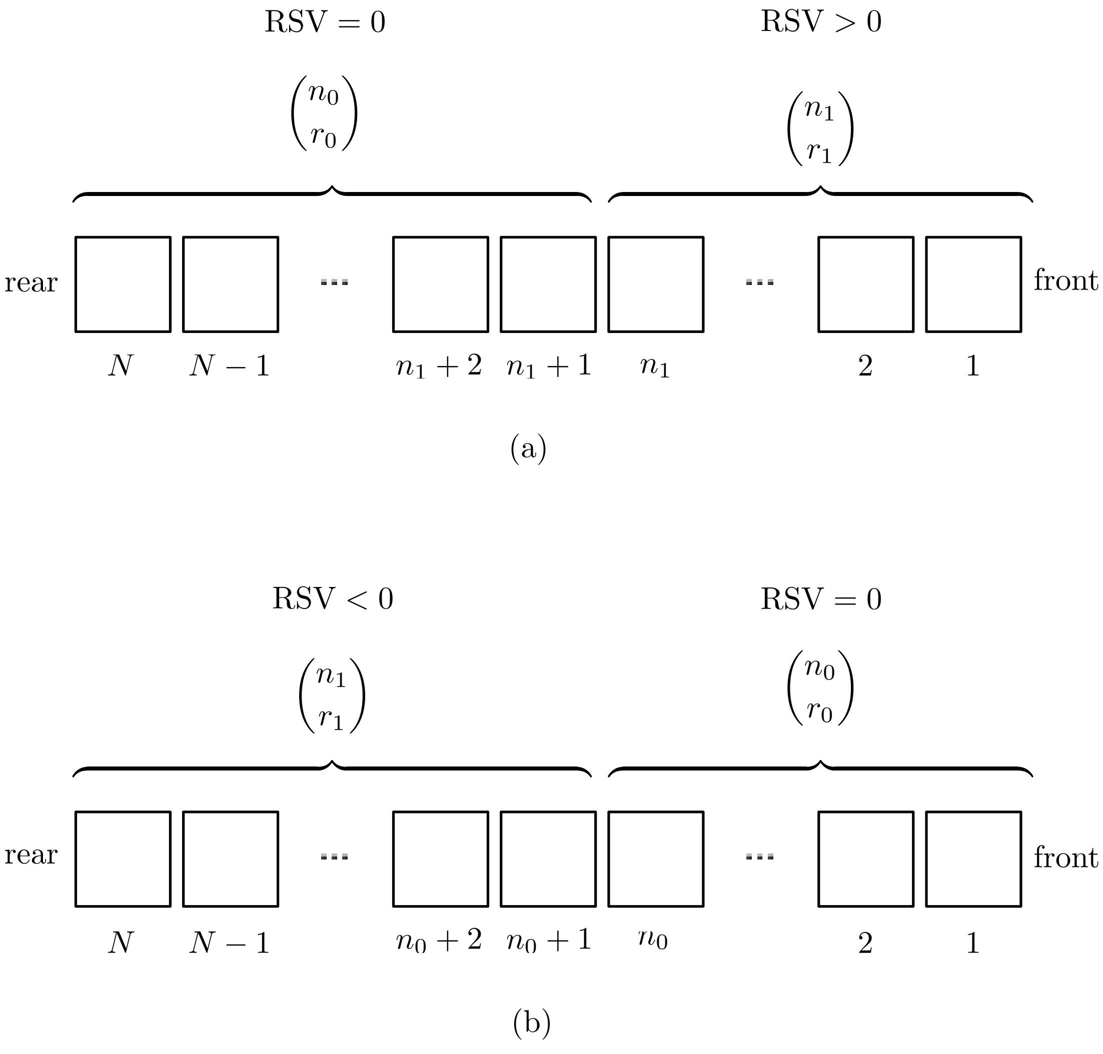


Figure 7.8: The Two General Ranking Possibilities.

Table 7.1: Feature Weights for Several Ranking Methods.

Ranking Method	Feature Weight
Best-case	$w = \log\left(\frac{p/(1-p)}{t/(1-t)}\right)$
Worst-case	$w = -\log\left(\frac{p/(1-p)}{t/(1-t)}\right)$
Random	$w = \begin{cases} \text{Best-case weight} & : \quad 1/2 \text{ of the time;} \\ \text{Worst-case weight} & : \quad 1/2 \text{ of the time.} \end{cases}$
Decision-theoretic	$w = \log\left(\frac{p/(1-p)}{q/(1-q)}\right)$
Inverse document frequency	$w = -\log(t)$
Coordination-level matching	$w = c$ (a positive constant)

		document term weight		
		< 0	0	> 0
query term weight	0	0	0	0
	1	< 0	0	> 0

Figure 7.9: RSVs and Their Relation to Query and Document Weights.

With this definition, the earlier equations for ASL and ASL$'$ can be rewritten, respectively, as

$$\text{ASL} = f_{\text{ASL}}(N, \mathcal{Q}, \mathcal{A})$$

and

$$\text{ASL}' = f_{\text{ASL}}(N, \mathcal{Q}', \mathcal{A}'). \tag{7.10.1}$$

These equations provide the best estimate of the Average Search Length for a query q and a ranking method rm; with parameters r_1, r_0, s_1, s_0, and N; when the quality of ranking argument to the f_{ASL} function is positive and the associated *document term weight* dtw_{rm} is also positive. Note that the quality of ranking measure is positive for all the ranking methods except for the worst-case ranking method, which has 0 as the value of its quality of ranking measure.

7.10.1 The ASL_{r}' Measure (a refined estimate of the Average Search Length)

A refined estimate of the Average Search Length for a query q, ranking method rm, and a weak 4-composition (r_1, s_0, r_0, s_1) can be obtained by taking the value of the quality of ranking method and the ranking method-specific document term weight into consideration. The evidence for this assertion comes from the information in Figure 7.8 on page 328. Notice that when the RSV is negative, all the documents that have feature frequency 1 are at the rear of the ranked sequence in Figure 7.8(b) on page 328 rather than being at the front as they are in Figure 7.8(a) on page 328. The implication of this observation is that the f_{ASL} function needs to be re-parameterized in some situations. Note that the quality of ranking value is positive for all of the ranking methods below, except for worst-case ranking, where it has a value of 0.

Ranking methods with positive $\mathcal{Q}'$ values always order documents with feature frequency 1 at the front of a ranked sequence of documents (shown by Figure 7.8(a)) on page 328 except when the document term weight for a weak 4-composition (r_1, s_0, r_0, s_1) is negative. In this case, the relative order of the document clusters are reversed and the situation in Figure 7.8(b) on page 328 occurs. To compensate for this possibility, the ASL value must be computed by the expression $f_{\mathrm{ASL}}(N, \mathcal{Q}, 1 - \mathcal{A})$, rather than by $f_{\mathrm{ASL}}(N, \mathcal{Q}, \mathcal{A})$, when the document term weight is negative.

The other situation to consider is the one in which the $\mathcal{Q}'$ value is 0. This only occurs for the worst-case ranking method. The behavior of this ranking method is the opposite of best-case ranking. Essentially, its Average Search Length computation has a behavior that is the opposite of its best-case counterpart. This means that when the document term weight for a weak 4-composition (r_1, s_0, r_0, s_1) is negative, the ASL value must be computed by the expression $f_{\mathrm{ASL}}(N, \mathcal{Q}, \mathcal{A})$. Otherwise, it must be computed by the expression $f_{\mathrm{ASL}}(N, \mathcal{Q}, 1 - \mathcal{A})$.

The Average Search Length measure that results from the possible re-parameterization is referred to as the $\mathrm{ASL}'_{\mathrm{r}}$ measure. Here is its description.

$$\mathrm{ASL}'_{\mathrm{r}} = \begin{cases} f_{\mathrm{ASL}}(N, \mathcal{Q}', \mathcal{A}'), & \text{if } (\mathcal{Q}' > 0 \text{ and } \mathrm{dtw}_{\mathrm{rm}} \geq 0) \text{ or } (\mathcal{Q}' = 0 \text{ and } \mathrm{dtw}_{\mathrm{rm}} < 0); \\ f_{\mathrm{ASL}}(N, \mathcal{Q}', 1 - \mathcal{A}'), & \text{if } (\mathcal{Q}' > 0 \text{ and } \mathrm{dtw}_{\mathrm{rm}} < 0) \text{ or } (\mathcal{Q}' = 0 \text{ and } \mathrm{dtw}_{\mathrm{rm}} \geq 0). \end{cases} \tag{7.10.2}$$

7.10.2 The $\mathrm{ASL}'_{\mathrm{g}}$ Measure (the gold standard for estimating the Average Search Length)

The value of the Average Search Length can also be obtained by mathematically modeling an actual ranking algorithm. The main objects of interest are the distributions of documents at the front and rear of a ranking, assuming, of course, that relevance is binary, that a term is either present or absent in a document, and that multiple occurrences of a term that is present have the same significance as just one occurrence of this term.

The $\mathrm{ASL}'_{\mathrm{g}}$ value (i.e., the gold standard ASL' value) is the ASL' value that can be obtained by performing the following actions. First, generate all the possible sequences of ranked documents for a query q, a document collection c of N documents, a ranking method *rm,* and parameters r_1, r_0, s_1, and s_0, where $r_1 + r_0 + s_1 + s_0 = N$, and N, plus each of the parameters, have values that are constrained to be natural numbers. Second, compute the total search length (TSL) for each of these sequences. Third, using the TSLs, compute the mean search length (MSL) for each sequence by dividing its TSL value by the number of relevant documents in the sequence. Note that each sequence has the same number of relevant documents. Finally, compute the $\mathrm{ASL}'_{\mathrm{g}}$ value by totaling the MSL values and then dividing that number by the number of sequences. The result of this is the $\mathrm{ASL}'_{\mathrm{g}}$ value for this query q, document collection, and ranking method. The $\mathrm{ASL}'_{\mathrm{g}}$ value for the other combinations of these entities can be obtained by the procedure that was just described in this paragraph.

The information in Table 7.2 on the following page is based on the information in Figure 7.8 on page 328 and Figure 7.9 on page 329. The notation n_{F}, n_{RF}, n_{R}, n_{RR} denotes the number of documents that are in the front cluster of a ranking, the number of relevant documents among these n_{F} front cluster documents, the number of documents that are in the rear cluster of a ranking, and the number of relevant documents among these n_{R} rear cluster documents, respectively. Note that the distributions for the second and third conditions in Table 7.2 on the next page are equivalent because when the condition $\mathrm{dtw}_{\mathrm{rm}} = 0$ holds, the retrieval status value is 0 for each document in a collection.

For a query, when every document has the same RSV, the calculation of $\mathrm{ASL}'_{\mathrm{g}}$ is greatly simplified because there is effectively only a single cluster. In this situation, we can pretend that either the front cluster does not exist (the second condition in Table 7.2) or that the rear cluster does not exist (the third condition in Table 7.2). These two conditions are equivalent. In general, when every document in a collection has the same RSV for a query, there is only a single cluster and all the documents are members of this cluster. The information in this paragraph is very important to the discussion in Section 8.6 (The Validation of $\mathrm{ASL}'_{\mathrm{g}}$).

Table 7.2: Document Distribution at the Front and Rear of An Actual Ranking.

condition	n_{R}	n_{RR}	n_{F}	n_{RF}
$\mathrm{dtw}_{\mathrm{rm}} > 0$	n_0	r_0	n_1	r_1
$\mathrm{dtw}_{\mathrm{rm}} = 0$	$n_1 + n_0$	$r_1 + r_0$	0	0
$\mathrm{dtw}_{\mathrm{rm}} = 0$	0	0	$n_1 + n_0$	$r_1 + r_0$
$\mathrm{dtw}_{\mathrm{rm}} < 0$	n_1	r_1	n_0	r_0

The Probability Generating Function Approach

We can use the results of Section 7.5 on page 298 to construct a probability generating function for $\mathrm{ASL}'_{\mathrm{g}}$. The ordinary generating function for the ranked documents that are at the front of the sequence is

$$\mathrm{FFfront}(x, z, n_{\mathrm{F}}) = \prod_{i=1}^{n_{\mathrm{F}}} (1 + x^i z).$$

The analogous ordinary generating function for the ranked documents that are at the rear of the sequence is

$$\mathrm{FFrear}(x, y, n_{\mathrm{F}}, N) = \prod_{i=n_{\mathrm{F}}+1}^{N} (1 + x^i y).$$

The ordinary generating function for the entire ordering, $G_2(x, y, z)$, is the convolution of the ordinary generating functions for the two parts of the ordering, namely, $\mathrm{FFfront}(x, z, , n_\mathrm{F})$ and $\mathrm{FFrear}(x, y, n_\mathrm{F}, N)$:

$$G_2(x, y, z, n_\mathrm{F}, N) = \mathrm{FFfront}(x, z, n_\mathrm{F}) \cdot \mathrm{FFrear}(x, y, n_\mathrm{F}, N).$$

Let

$$F(x) = G_2(x, y, z, n_\mathrm{F}, N)|_{y=1,z=1}$$

be the expression that is obtained from the expansion of G_2 when the value 1 is substituted everywhere that a y or z appears in the expanded form. This resultant expression, denoted by $F(x)$, is now a function of just one variable, namely, x, because, for a given query, the values of $n_\mathrm{F}, n_\mathrm{RF}, n_\mathrm{R}, n_\mathrm{RR}$, and N can be treated as constants.

The probability generating function, $\mathrm{PGF}(x)$, for ASL'_g can be defined as

$$\mathrm{PGF}(x, n_\mathrm{F}, n_\mathrm{RF}, n_\mathrm{R}, n_\mathrm{RR}, N) = M(x) / \left(\binom{\mathrm{n_F}}{\mathrm{n_{RF}}} \binom{\mathrm{n_R}}{\mathrm{n_{RR}}} \right),$$

under the assumption that each of the $\binom{\mathrm{n_F}}{\mathrm{n_{RF}}}\binom{\mathrm{n_R}}{\mathrm{n_{RR}}}$ possible orderings is equally likely and $M(x)$ is the result of adapting the $F(x)$ equation to take into account that each ordering has $n_\mathrm{RR} + n_\mathrm{RF}$ relevant documents. The adaptation involves dividing each exponent (which represents the total search length of an ordering) in this function by this number of relevant documents. For example, if

$$F(x) = x^7 + 2x^8 + 3x^9 + 3x^{10} + 3x^{11} + 2x^{12} + x^{13}$$

and the number of relevant documents in each ordering is 3, then

$$M(x) = x^{7/3} + 2x^{8/3} + 3x^{9/3} + 3x^{10/3} + 3x^{11/3} + 2x^{12/3} + x^{13/3},$$

is the ordinary generating function for the random variable X. From this, $\mathrm{ASL}'_{\mathrm{g}}$ can be calculated by taking PGF', the first derivative of PGF with respect to x, and evaluating the resultant expression at $x = 1$. That is,

$$\mathrm{ASL}'_{\mathrm{g}} = \mathrm{PGF}'(x, n_{\mathrm{F}}, n_{\mathrm{RF}}, n_{\mathrm{R}}, n_{\mathrm{RR}}, N)\big|_{x=1.} \qquad (7.10.3)$$

The Combinatoric Approach

This approach, like the prior one, also makes use of the information in Table 7.2 on page 333. In addition, it makes use of the result from Lemma 7.2.2 on page 276. Subsequent discussions show that the significant difference between these two approaches to calculating $\mathrm{ASL}'_{\mathrm{g}}$ is that the latter approach is extremely efficient computationally for large values of N. The advantage of the generating function approach is that it has much to offer if there is interest in also investigating higher order moments (e.g., variance, kurtosis, skewness) of the search length function around a constant c. The $\mathrm{ASL}'_{\mathrm{g}}$ value is based on information from the first moment around 0 (i.e., the mean). If this is *the* only moment that one is interested in, then there is no need to use the probability generating function approach as this second approach is much, much more computationally- and memory-efficient than the former approach.

The first step in the derivation of $\mathrm{ASL}'_{\mathrm{g}}$ with this combinatoric approach is to treat the front and rear RSV clusters in a ranking as independent. To effect this, we develop situation-specific equations for these clusters and mathematically combine them to provide an equation for $\mathrm{ASL}'_{\mathrm{g}}$. Since N, the number of documents in a collection, is assumed to be at least 1, the number of RSV clusters for a collection is either 1 or 2. It is 1 only when all the documents have the same RSV. This occurs only when $\mathrm{n}_{\mathrm{R}} = 0$ is true or $n_{\mathrm{F}} = 0$ is true. In all other situations, there are two clusters. The next step is to determine the total unnormalized search lengths for each of these three cases. Finally, these results are combined and scaled by the total number of relevant documents to give $\mathrm{ASL}'_{\mathrm{g}}$.

Lemma 7.10.1. *The Average Search Length equation for this approach is*

$$\mathrm{ASL}'_{\mathrm{g}} = \begin{cases} \dfrac{N+1}{2}, & \textit{if there is exactly one cluster;} \\ \dfrac{\mathrm{MSL}_{\mathrm{gold,front}} + \mathrm{MSL}_{\mathrm{gold,rear}}}{n_{\mathrm{RF}} + n_{\mathrm{RR}}}, & \textit{otherwise;} \end{cases}$$

where

$$\begin{aligned} \mathrm{MSL}_{\mathrm{gold,front}} &= \frac{n_{\mathrm{RF}}(n_{\mathrm{F}}+1)}{2}, \\ \mathrm{MSL}_{\mathrm{gold,rear}} &= \frac{n_{\mathrm{RR}}(\mathrm{n}_{\mathrm{R}}+1)}{2} + \mathrm{shift\_contribution}, \ \textit{and} \\ \mathrm{shift\_contribution} &= n_{\mathrm{RR}} \cdot n_{\mathrm{F}}. \end{aligned}$$

Proof. The proof is by cases.

There is exactly one cluster (i.e., $\mathrm{n}_{\mathrm{F}} = 0$ *or* $\mathrm{n}_{\mathrm{R}} = 0$, *but not both).*
This means that either $n_{\mathrm{F}} = 0$ is true or $\mathrm{n}_{\mathrm{R}} = 0$ is true. These conditions cannot be simultaneously true because the computation of the Average Search Length assumes that there is at least one relevant document in a collection for a query. Effectively, this is a special case because the front and the rear clusters are identical in this situation. Without loss of generality, let $N = \mathrm{n}_{\mathrm{R}} + n_{\mathrm{F}}$ be the total number of documents in the collection for a query q and let $R = n_{\mathrm{RR}} + n_{\mathrm{RF}}$ be the total number of relevant documents in the collection for q.

The document positions range from 1 to N, inclusive. Each of the n_{RF} relevant documents has the same probability of occupying any of these positions. Only one document can occupy a position at a time. All documents must occupy one position.

By Lemma 7.2.2 on page 276, each of the N positions occurs exactly $\frac{R\binom{N}{R}}{N}$ times in the sample space of R-combinations for these N positions. The weights of these positions are simply their position values. The sum of these weights is

$$\sum_{i=1}^{N} i = \binom{N+1}{2}.$$

Multiplying this sum by the frequencies of the positions yields

$$\frac{R\binom{N}{R}}{N}\binom{N+1}{2},$$

the total weight of the positions occupied by the relevant documents in the sample space. The $\mathrm{MSL_g}$ value is this total weight divided by the cardinality of the sample space, that is,

$$\begin{aligned}\mathrm{MSL_g} &= \frac{\frac{R\binom{N}{R}}{N}\binom{N+1}{2}}{\binom{N}{R}} \\ &= \frac{R}{N}\binom{N+1}{2} \\ &= \frac{R}{N}\frac{(N+1)N}{2} \\ &= \frac{R(N+1)}{2}.\end{aligned}$$

Similarly, the $\mathrm{ASL'_g}$ value is the $\mathrm{MSL_g}$ value divided by R, the number of relevant documents, that is,

$$\begin{aligned}\mathrm{ASL'_g} &= \mathrm{MSL_g}/R \\ &= \frac{R(N+1)}{2\ R} \\ &= \frac{N+1}{2}.\end{aligned}$$

There are two clusters (i.e., $\mathrm{n_R} > 0$ *and* $n_\mathrm{F} > 0$*).*

From the results of the previous case, the MSL equation for the front cluster can be obtained from the the $\mathrm{MSL}_{\mathrm{exact}}$ equation by substituting n_{RF} for R and n_{F} for N everywhere that they occur in this equation. Therefore, we have

$$\mathrm{MSL}_{\mathrm{gold,front}} = \frac{n_{\mathrm{RF}}(n_{\mathrm{F}}+1)}{2}.$$

The MSL value for the rear cluster was obtained in a similar manner. Its equation is

$$\mathrm{MSL}_{\mathrm{gold,rear}} = \frac{n_{\mathrm{RR}}(\mathrm{n}_{\mathrm{R}}+1)}{2} + \mathrm{shift\_contribution},$$

where

$$\mathrm{shift\_contribution} = n_{\mathrm{RR}} \cdot n_{\mathrm{F}}$$

and represents the contribution to the MSL that the n_{RR} relevant documents make. The n_{F} part of the shift_contribution equation represents the number of positions that the first position in the rear cluster is from the first position in the front cluster. The Average Search Length is then calculated by

$$\mathrm{ASL}_{\mathrm{g}}' = \frac{\mathrm{MSL}_{\mathrm{gold,rear}} + \mathrm{MSL}_{\mathrm{gold,front}}}{n_{\mathrm{RF}} + n_{\mathrm{RR}}}.$$

□

7.11 Summary

The main contributions of this chapter were the development of combinatorial models of the unnormalized (ASL) and normalized ($\mathcal{A}$) search lengths, a proof that the probabilistic and combinatorial formulas for $\mathcal{A}$ were equivalent, a demonstration of how Gaussian polynomials could be used to develop a combinatoric-based formula for the ASL, how Gaussian polynomials could be used to provide detailed distributional information on the sums of the positions of the relevant documents in an optimal ranking, the development of formulas for the expected value and variance of the ASL and $\mathcal{A}$, and the development of a family of ASL measures (i.e., $\mathrm{ASL}', \mathrm{ASL}'_{\mathrm{r}}, \mathrm{ASL}'_{\mathrm{g}}$) that could be weakly-ordered. The main result of this weak order was that it is possible to state that a particular measure is either at least as accurate, or is at most as accurate, as any of the other two ASL variants in this family.

Chapter 8

Validation of the Formulas for the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL' Measures

In the past several chapters, various combinatoric-based formulas were developed to help calculate the $\mathcal{Q}'$ (quality of ranking) values for the coordination level matching (CLM), inverse document frequency (IDF), and decision-theoretic (DT) ranking methods; to calculate the value of the $\mathcal{A}'$ (normalized search length) measure; and to calculate the value of the ASL' (Average Search Length) measure. The $\mathcal{Q}'$, $\mathcal{A}'$, and ASL' entities, respectively, are special analogs of the $\mathcal{Q}$, $\mathcal{A}$, and ASL measures. These special analogs have been defined so that singularities cannot occur when they are used to calculate values for the $\mathcal{Q}$, $\mathcal{A}$, and ASL entities. In the great majority of cases, they calculate exactly the same values as their $\mathcal{Q}$, $\mathcal{A}$, and ASL counterparts. The details of how the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL' adaptations of these counterparts were developed can be found in Section 4.3, which starts on page 117. Various mathematical arguments were used to prove the validity of these formulas. This section details additional methods that were used to provide further confidence in the formulas.

The process of validating the value of a particular measure, say $\mathcal{Q}'$, by more than one method, is an example of a "valuable and widely used strategy [known as *triangulation*] which involves the uses of multiple sources to enhance the rigour of ... research" (Robson, 2002). The common theme throughout this chapter is the validation of measures such as $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ by multiple methods.

The random variables $\mathcal{B}$ (quality of ranking for a measure), $\ddot{\mathcal{M}}$ (normalized search length) and $\mathcal{L}$ (unnormalized search length) were introduced and defined in previous chapters and are key components of the validation process. In those chapters, formulas for the mean and variance were derived for each of these random variables. The expected values of these random variables are the $\mathcal{Q}'$, $\mathcal{A}'$, and ASL$'$ measures, respectively. In this validation process, the expected value and variance of each of these random variables were calculated in two ways: by the formulas developed in earlier chapters and by statistical methods and specially-developed software that obtained their input from specially-constructed test datasets. Combinatorial generation and enumeration algorithms were used to help populate these datasets.

In the last section of this chapter, we also discuss the estimation of $\mathcal{Q}'$ values by random sampling for the CLM and IDF ranking methods. In that section, we use well-known statistical tests to determine if there is any significant difference between the estimated $\mathcal{Q}'$ values and the exact ones that were calculated by the techniques that were developed in the previous chapters. During the validation process that is described in this chapter, the efficient calculation of $\mathcal{Q}'$ values became increasingly important because, as the number of documents in a collection increased, the exponential behavior of the algorithms caused them to require exponential amounts of computational and memory resources. This resulted in situations where the computation of $\mathcal{Q}'$ values, even for N (the number of documents in a collection) in the low hundreds, became infeasible. In practice, it has been found that the efficient calculation of $\mathcal{Q}'$ values was not much of a concern for small (e.g., $1 \leq N \leq 200$) document collections but that it rapidly became a concern as the value of N grew.

The validation that is discussed in this chapter, and the mathematical work that took place in the prior chapters, plus the work that was discussed in subsequent chapters, was accomplished primarily by the use of Mathematica® (Wolfram, 2003) programs. The other software and languages used include the statistical programming language R (Chambers, 2008; Dalgaard, 2008; Rizzo, 2008; Spector, 2008), the mathematical statistics package mathStatica (Rose and Smith, 2002), the general programming language C (Harbison and Steele, 2002), and the general object-oriented programming language C++ (Stroustrup, 2000).

Mathematica® (Wolfram, 2003) is a computer algebra system that has many strengths in the area of symbolic computation. It is used mainly in scientific and mathematical fields, engineering, and technical computing. The use of Mathematica® (Wolfram, 2003) in the work for this dissertation helped to eliminate the tedium that is often associated with many of the computations that were performed. The work for this dissertation made use of its many mathematical functions and its support in these topical areas: calculus, polynomial algebra, number theory, numbers and precision, equation solving, statistics, and discrete mathematics.

The R programming language (Chambers, 2008; Dalgaard, 2008; Rizzo, 2008; Spector, 2008), is a popular, free, open source statistical programming language and environment for statistical computing and graphics. In this dissertation, R was used to help analyze several kinds of results by applying the Kolmogorov-Smirnov (Conover, 1999) and Wilcoxon signed ranks (Conover, 1999) tests to various datasets.

The mathStatica toolset was developed by Colin Rose and Murray Smith for doing work in mathematical statistics (Rose and Smith, 2002), it is an add-on to Mathematica®. It was used to help corroborate the expected value and variance equations for the $\mathcal{Q}$ measures that appeared in Chapters 5 and 6.

C (Harbison and Steele, 2002) is a general purpose programming language and C++ is the object-oriented version of the C programming language. The algorithms that were used for lexical analysis, stoplists, and stemming came from code in Frakes and Baeza-Yates (1992) that was written in C and, therefore, could not be directly used by Mathematica®. The solution was to use the Mathematica® Mathlink API for C in order to communicate with the compiled version of the code from Frakes and Baeza-Yates (1992). In order to accomplish this, though, the C code had to undergo moderate modifications so that it would fit into the Mathematica® Mathlink API framework. The author of this dissertation considered this choice to be a much better choice than trying to reimplement the lexical analysis, stoplists, and stemming code entirely in Mathematica®. C++ programs were used to help with creating correct XML (Stanek, 2002) versions of the Cystic Fibrosis datasets. Some of the XML versions of the Cystic Fibrosis datasets that were downloaded from the World Wide Web (Berners-Lee and Fischetti, 1999) did not pass validation by Mathematica's XML parser. Upon inspection of these datasets, it was found that parts of these them were not well-formed with respect to the XML language. Other problems were also found, including that sometimes parts of what should have been two entries had been merged into one entry.

8.1 The Validation of $\mathcal{Q}'$

8.1.1 Test Data Generation

This section discusses the creation of three datasets, namely, the $\texttt{NAT01}^{(\mathrm{N})}$, **cgeRVN**, and **analyticRVN** datasets. Neither of these datasets was an IR dataset. That is, they did not contain entities such as queries, documents, or relevance judgments. Instead, these datasets contained values that could be analyzed by statistical software, such as R, if necessary, to determine whether the analytically-determined values for $\mathcal{Q}'$ were exactly equal to the empirically-determined versions of these values. More details are contained in the subsequent paragraphs.

The first stage in the validation of the formulas for $\mathcal{Q}'_{\text{CLM}}$, $\mathcal{Q}'_{\text{IDF}}$, and $\mathcal{Q}'_{\text{DT}}$; for a collection of $1 \leq N \leq 200$ documents with parameters r_1, s_0, r_0, and s_1; created a dataset $\texttt{NAT01}^{(\text{N})}$ that provided the data to calculate the random variables $\mathcal{B}_{\text{CLM}}$, $\mathcal{B}_{\text{IDF}}$, and $\mathcal{B}_{\text{DT}}$. The $\texttt{NAT01}^{(\text{N})}$ dataset has $\binom{N+3}{3}$ rows and three columns (e.g., nat01_CLM, nat01_IDF, and nat01_DT) that corresponded respectively, to the random variables just mentioned. This dataset was populated by visiting each member of the set of weak 4-compositions for N (denoted by $W4C_N$), in turn, and performing the following actions for each member: (1) construct a new row for the $\texttt{NAT01}^{(\text{N})}$ dataset; (2) set the value for the column corresponding to a ranking method to 1 if the member meets the qualification criteria for this method; otherwise, set the value to 0; and (3) insert this row into the dataset. For example, assume that the ranking method-specific criteria is met for the CLM and IDF methods but not for the DT method. Then, the row that is inserted into the dataset had a value of 1 for columns nat01_CLM and nat01_IDF, but had a value of 0 for column nat01_DT. At the conclusion of the visitation process, the number of rows in the $\texttt{NAT01}^{(\text{N})}$ dataset was equal to the cardinality of set W4C_N. Note that, if the value of N was, say 5, then the name of the dataset was $\texttt{NAT01}^{(5)}$ and that the name of the set of weak 4-compositions for that value of N was $W4C_5$.

For the convenience of the reader, the decision criteria mentioned in (2) and that appeared in the Mean and Variance sections of the chapters for the $\mathcal{Q}_{\text{CLM}}$, $\mathcal{Q}_{\text{IDF}}$, and $\mathcal{Q}_{\text{CLM}}$ ranking methods are repeated here. For an arbitrary member x of the set of weak 4-compositions for N,

$$\begin{aligned}
\mathcal{B}_x^{\text{CLM}} &= \text{BooleToNat}(p'_x > t'_x), \\
\mathcal{B}_x^{\text{IDF}} &= \text{BooleToNat}((p'_x > t'_x) \textbf{ or } ((p'_x \leq t'_x) \textbf{ and } (t'_x = 1 - \epsilon))), \text{ and} \\
\mathcal{B}_x^{\text{DT}} &= \text{BooleToNat}((p'_x > \max(t'_x, q'_x)) \textbf{ or } (p'_x \leq \min(t'_x, q'_x))),
\end{aligned}$$

where x is a weak 4-composition of N; p'_x, t'_x, and q'_x are the p', t', and q' values, respectively, for that x; and

$$\epsilon = \begin{cases} N^{-2}, & \text{if } N \geq 2; \\ 10^{-4}, & \text{otherwise.} \end{cases}$$

The second stage processed the data in the `NAT01`$^{(\mathrm{N})}$ dataset to generate the expected value and population variance associated with each of its columns. The expected value for a column was synonymous with the $\mathcal{Q}'$ value for the ranking method that corresponds to that column. These generated quantities were placed in a new dataset named `cgeEVN` (created by combinatorial generation and enumeration (cge)). This dataset had 10 columns:

N,
E_nat01_CLM, Var_nat01_CLM, nqc_nat01_CLM,
E_nat01_IDF, Var_nat01_IDF, nqc_nat01_IDF,
E_nat01_DT, Var_nat01_DT, and nqc_nat01_DT.

The first column, N, represented the number of documents in the collection; the second (nat01_IDF), fifth (E_nat01_IDF), and eighth (E_nat01_DT) columns represented the expected values for the coordination level matching, inverse document frequency, and decision-theoretic ranking methods, respectively; the third, sixth, and ninth columns represented the population variances for these methods; and the fourth, seventh, and tenth columns represented the number of weak 4-compositions that met the qualification criteria for the respective methods for a collection of N documents. This dataset had exactly one row for each value of N.

The third stage in the process of data generation for these validation purposes was to construct a dataset, named `analyticEVN`, that had the same columns as those that were in the `cgeEVN` dataset. The major difference was in the provenance of the values: the values in this new dataset were derived from the formulas in Chapters 4–7, inclusive, and the Principle of Inclusion-Exclusion (for the CLM and IDF ranking methods), rather than by combinatorial generation and enumeration. For each value of N, these values were compared eventually to their counterparts in the former dataset. For example, if $N = 5$, the values in the row of the latter dataset, where $N = 5$, were compared to their counterparts in the latter dataset where $N = 5$.

These first two stages (Technique 1) contrasted sharply with the third one (Technique 2) with respect to the generation of expected values and population variances: combinatorial generation and enumeration were used for the first two stages whereas analytic formulas, in conjunction with the Principle of Inclusion and Exclusion, were the mainstay of the final stage. Technique 1 was inefficient because the cardinality of set $\mathrm{W4C_N}$ was $\Theta(N^3)$ and, even though many of its members had no possibility of being a qualifying weak 4-composition, they still had to be generated and examined; Technique 2 was much more efficient because it used solely closed-form expressions for the DT ranking method in conjunction with the concept of divisor pairs to quickly eliminate most of the non-qualifying weak 4-compositions at the *beginning* of the data generation process. The Principle of Inclusion-Exclusion was used later to exclude any 4-compositions that satisfied the qualification criteria but that are duplicates of other composition that also satisfied the same criteria.

The `cgeRVN`and `analyticRVN` datasets had 200 rows each. In each dataset, the values for the column named `N` ranged from 1 to 200, inclusive. The rows represented document collections that ranged in cardinality from 1 to 200 documents. There were two reasons that the values were in this range. The more important one was that, except for the boundary cases (i.e., $N = 0$ and $N = 1$), the formulas for Technique (2) were based on number-theoretic properties rather than specific values of N. The other reason was that, due to the problem of combinatorial explosion, it was prudent, during testing, to restrict N to a moderate size. This was a large problem for Technique 1 because the cardinality of W4C$^{(\mathrm{N})}$ was a cubic function of N. It could also be a problem for Technique 2 because the number of subsets of divisor pairs that had to be intersected was $\Theta(2^{\tau(\mathrm{N})+1})$, where $\tau(N)$ was the number of divisor pairs for N. The big-theta notation expression came from the fact that, since the cardinality of the superset for a set of i divisor pairs was 2^i, the total number of subsets for a collection of N documents was

$$\sum_{i=1}^{\tau(N)} 2^i = 2^{\tau(N)+1} - 2.$$

Table E.1 on page 544, Table E.2 on page 545, and Table E.3 on page 546 in Appendix E list the $\mathsf{nqc\_nat01\_CLM}$, $\mathsf{nqc\_nat01\_IDF}$, and $\mathsf{nqc\_nat01\_DT}$ values for document collections that range in size from 1 to 200 documents, inclusive. Given an $\mathsf{nqc\_}rm$ value for a document collection with ranking method rm and size $N \geq 1$, the corresponding $\mathsf{E\_nat01\_}rm$ and $\mathsf{Var\_nat01\_}rm$ values can be obtained in a straightforward way by the following two transformations:

$$\begin{aligned}\mathsf{E\_nat01\_}rm &= \frac{\mathsf{nqc\_nat01\_}rm}{\binom{N+3}{3}}\\ \mathsf{Var\_nat01\_}rm &= \mathsf{E\_nat01\_}rm(1 - \mathsf{E\_nat01\_}rm).\end{aligned}$$

8.1.2 Empirical Data Supports the Validation of $\mathcal{Q}'$

Software was developed to compare the 200 observations in the datasets. For each row in the **cgeRVN** dataset, identified by a value of v for its column named N, the values for the remaining columns in that row were compared to their counterparts in the row of the **analyticRVN** dataset that also had v for the value of its column that was named N. The values in corresponding rows of the datasets were found to be equal, that is, the two techniques generated identical datasets. Therefore, it was concluded that the equations that were developed in Chapter 4, Chapter 5, and Section 6.1 through Section 6.3, inclusive, to compute the ranking method-specific $\mathcal{Q}$ values, are valid.

8.2 The Validation of $\mathcal{Q}'$ Estimates That Were Obtained by Random Sampling

Table 8.1 on the following page shows the minimum sample sizes for estimating $\mathcal{Q}$ and $\mathcal{Q}'$ values with a margin of error of either .01 or .05 for several document collection sizes. The sample size calculation formula that was used in this dissertation was the one from Levy and Lemeshow (2008):

$$n \geq \frac{z^2 N V_x^2}{z^2 V_x^2 + (N-1)\epsilon^2}, \tag{8.2.1}$$

where n is the minimum sample size, ϵ is the margin of error, z is the critical value for a $1 - \epsilon$ confidence interval ($z = 1.95996$ for a 95% confidence interval; $z = 2.57583$ for a 99% confidence interval), N is the population size, and V_x^2 is the estimate of the variance.

The true variance is generally unknown and often must be estimated from the range R of possible values. For $\mathcal{Q}$, the lower bound of this range was 0 and the upper bound was 1, thereby giving a range of $R = 4 - 0 = 4$. A common technique that is used to estimate the sample variance is the $R/4$ method (Mendenhall et al., 1971; Browne, 2001; Hozo et al., 2005). With this method, the sample variance estimate is the range divided by 4. For $\mathcal{Q}$, this meant that

$$\begin{aligned} V_x^2 &= R/4 \\ &= (1-0)/4 \\ &= 1/4. \end{aligned}$$

With this result, Inequality 8.2.1 on the previous page simplifies to

$$n \geq \left\lceil \frac{z^2 N/4}{z^2/4 + (N-1)\epsilon^2} \right\rceil . \tag{8.2.2}$$

Table 8.1: Minimum Sample Sizes for Estimating $\mathcal{Q}$ With the Specified Margin of Error.

N	# of weak 4-comps	ME = .01	ME = .05
1	4	4	4
5	56	56	49
10^1	286	282	165
10^2	176851	15165	384
10^3	167668501	16586	385
10^4	166766685001	16588	385
10^5	166676666850001	16588	385
10^6	166667666668500001	16588	385
10^7	166666766666685000001	16588	385
10^8	166666676666666850000001	16588	385
10^9	166666667666666668500000001	16588	385
10^{10}	166666666766666666685000000001	16588	385

8.2.1 Test Data Generation

Inequality 8.2.2 was used to determine the minimum sample sizes for document collections with sizes that ranged from 1 to 200, inclusive. These sizes were for two margins of error (i.e., .01 and .05) and were used to estimate the $\mathcal{Q}'_{\text{CLM}}$ and $\mathcal{Q}'_{\text{IDF}}$ values for all 200 collection ranges. A dataset was created to store these values. Since these values are not queries, documents, or relevance judgments, this dataset is not an IR dataset. This dataset had 7 columns:

N,
q_CLMgold, q_CLM01, q_CLM05,
q_IDFgold, q_IDF01, and q_IDF05.

This dataset contained one row for each distinct value of N, for a total of 200 rows.

The first column, N, of a row represented the number of documents in a collection. The second column represented the *exact* (i.e., actual) $\mathcal{Q}'_{\text{CLM}}$ value. The third and fourth columns represented $\mathcal{Q}'_{\text{CLM}}$ values that were *estimated* from random samples with .01 and .05 margins of error, respectively. The remaining three columns were the IDF counterparts of the second, third, and fourth columns, respectively.

8.2.2 Empirical Data Supports the Validation of $\mathcal{Q}'$ Estimates That Were Obtained by Random Sampling

The Wilcoxon signed ranks test (with continuity correction) (Conover, 1999) was run on selected columns to determine if there was any significant difference between the means of the exact $\mathcal{Q}'$ values and their associated means of the estimated values. The 4 hypotheses and their significance levels are listed in Table 8.2 on the next page, along with the p-values that were computed by the Wilcoxon signed ranks tests. Each of the p-values were large and indicated that the differences in the means being compared were well within the differences expected under the null hypotheses. Therefore, there was no reason to suspect that the null hypotheses were false.

The practical consequence of these results was that, based on collection size of less than or equal to 200 documents, random sampling with a .05 margin of error sufficed for $\mathcal{Q}'$ estimation. If more confidence in the estimated value was required, or desired, then the $\mathcal{Q}'$ values could be estimated with a smaller margin of error.

8.3 The Validation of $\mathcal{A}'$

The strategy for validating $\mathcal{A}'$, for a document collection of size $N \geq 1$, were guided by the sets of relationships that are enumerated in Table 8.3 on page 353 and in Table 8.4

Table 8.2: Wilcoxon signed ranks test with continuity correction ($\alpha = 0.01$, two-tailed).

H_0 : q_CLMgold = q_CLM01	H_0 : q_CLMgold = q_CLM05
H_1 : q_CLMgold $\neq$ q_CLM01	H_1 : q_CLMgold $\neq$ q_CLM05
$p-$value = 0.7215	$p-$value = 0.1607
action: fail to reject the null hypothesis	action: fail to reject the null hypothesis
H_0 : q_IDFgold = q_IDF01	H_0 : q_IDFgold = q_IDF05
H_1 : q_IDFgold $\neq$ q_IDF01	H_1 : q_IDFgold $\neq$ q_IDF05
$p-$value = 0.7273	$p-$value = 0.2227
action: fail to reject the null hypothesis	action: fail to reject the null hypothesis

on page 353 for various values of p and t. Jointly, these values defined 12 categories, each with a set of three relationships between pairs of variable values (i.e., p and t, p' and t', $\mathcal{A}$ and $\mathcal{A}'$). Table 8.3 on page 353 enumerates the 9 categories of relationships that exist when a collection contains at least one relevant document and Table 8.4 on page 353 enumerates the three categories of relationships that exist for a collection that does not have any relevant documents.

The strategy for validating $\mathcal{A}'$ consisted of both exhaustive and selective checking of the sets of three analytically-determined conditions that were associated with the 12 categories across different ranges of collection cardinalities. For a particular value of $N \geq 1$ and a weak 4-composition w for that value, w was a member of exactly one of these 12 categories.

The exhaustive checking involved the enumeration of all the weak 4-compositions for $1 \leq N \leq 200$, determining which of the 12 categories each weak composition was a member of, and then determining if the set of three relationships for that category held for this weak composition. Similarly, the selective checking involved the enumeration of all the weak 4-compositions for $201 \leq N \leq 400$, determining which of the 11 (instead of 12) categories each weak composition was a member of, and then determining if the set of three relationships for that category held for this weak composition. The excluded category was the one in Table 8.3 on the following page where the joint conditions $0 < p < 1$ and $0 < t < 1$ hold. The main reasoning behind excluding this category was combinatorial explosion and is discussed in greater detail in Section 8.3.3. Essentially, the selective checking only involved the verification of what can be considered boundary conditions (i.e., p is either undefined or has the value 0 or 1 whereas t has the value 0 or 1). This is discussed further in Section 8.3.1.

The general conditions for the sameness, or difference, of the $\mathcal{A}$ and $\mathcal{A}'$ values in Table 8.3 on the next page and Table 8.4 on the following page are enumerated in the first two lines of each cell in the square matrix that is depicted in Table 8.3 on the next page. The third line of each cell in this table shows that the values for the $\mathcal{A}$ and $\mathcal{A}'$ measures are the same for the joint conditions on the main diagonal of the matrix because

$$-p + t = -p' + t'$$

holds for each of the 3 cells there. These conditions do not hold in the other 6 cells of the figure.

Note that the $\mathcal{A}$ measure is only defined for document collections that have at least one relevant document. By contrast, the analogous measure, $\mathcal{A}'$, is defined for all document collections that are parameterized by the variables r_1, s_0, r_0, and s_1, even if the collection does not have any relevant documents for a particular query q.

8.3.1 Boundary Conditions

There were only 5 boundary conditions that had to be considered in the validation efforts for $\mathcal{A}'$. Therefore, the only members of the set of weak 4-compositions that needed to

Table 8.3: The relationships between $p, t, p', t', \mathcal{A}$, and $\mathcal{A}'$ when a collection has at least one relevant document (both $\mathcal{A}$ and $\mathcal{A}'$ are defined for each of the 9 categories).

	$t = 0$	$0 < t < 1$	$t = 1$
$p = 0$	$p = t = 0$ $p' = t' = \epsilon$ $A = A'$	$p = 0, p' = \epsilon$ $t = t'$ $A \neq A'$	$p = 0, t = 1$ $p' = \epsilon, t' = 1 - \epsilon$ $A \neq A'$
$0 < p < 1$	$p = p', t = 0$ $t' = \epsilon$ $A \neq A'$	$p = p'$ $t = t'$ $A = A'$	$p = p', t = 1$ $t' = 1 - \epsilon$ $A \neq A'$
$p = 1$	$p = 1, t = 0$ $p' = 1 - \epsilon, t' = \epsilon$ $A \neq A'$	$p = 1, t = t'$ $p' = 1 - \epsilon$ $A \neq A'$	$p = t = 1$ $p' = t' = 1 - \epsilon$ $A = A'$

Table 8.4: The relationships between $p, t, p', t', \mathcal{A}$, and $\mathcal{A}'$ when a collection does not have any relevant documents ($\mathcal{A}'$ is defined, but $\mathcal{A}$ is undefined for each of the three categories).

	$t = 0$	$0 < t < 1$	$t = 1$
p is undefined	$p' = t' = \epsilon, t = 0$ A is undefined $A' = 1/2$	$p' = \epsilon, t = t'$ A is undefined $A' = (1 - \epsilon + t')/2$	$p' = \epsilon, t = 1, t' = 1 - \epsilon$ A is undefined $A' = 1 - \epsilon$

be examined were those where at least one of the following 5 conditions was true: (1) p was undefined, (2) $p = 0$, (3) $t = 0$, (4) $p = 1$, or (5) $t = 1$. These conditions were associated with Scenario 1 (none of the documents were relevant), Scenario 2 (none of the relevant documents contained the query term), Scenario 3 (none of the documents contained the query term), Scenario 4 (all the relevant documents contained the query term), and Scenario 5 (all documents contained the query term) below, respectively. Table 8.5 enumerates them and other pertinent information.

The $\mathcal{A}$ measure possibly differs from its $\mathcal{A}'$ counterpart only in 5 scenarios (see Table 8.5). In each of these scenarios, the sum $r_1 + s_0 + r_0 + s_1$ equals N. The $\star$ (star) in this table represents an integer value in the closed interval $[0, N]$ whereas the + (plus) represents an integer value in the closed interval $[1, N]$. The symbol ϵ represents a small value that is close to 0 and is a value that can never occur in the range of possible values for p or t when $N \geq 1$. Its value is N^{-2} when $N \geq 2$ and is 10^{-4} otherwise.

Table 8.5: Special Scenarios for $\mathcal{A}$ and $\mathcal{A}'$ (Before Subsumption).

Scenario	r_1	s_0	r_0	s_1	Comment	# weak 4-comps	$\mathcal{A}$	$\mathcal{A}'$
1	0	$\star$	0	$\star$	p is undefined, $p' = \epsilon$	$N+1$	undefined	$1-\epsilon$
2	0	$\star$	$\star$	$\star$	$p = 0$, $p' = \epsilon$	$\binom{N+2}{2}$	$\frac{1+t}{2}$	$\frac{1-\epsilon+t'}{2}$
3	0	$\star$	$\star$	0	$t = 0$, $t' = \epsilon$	$N+1$	$\frac{1-p}{2}$	$\frac{1-p+\epsilon}{2}$
4	+	$\star$	0	$\star$	$p = 1$, $p' = 1-\epsilon$	$\binom{N+1}{2}$	$\frac{t}{2}$	$\frac{\epsilon+t'}{2}$
5	+	0	0	+	$t = 1$, $t' = 1-\epsilon$	$N-1$	$1-\frac{p}{2}$	$1-\frac{p'+\epsilon}{2}$

Scenario 1: None of the documents were relevant

The value of $r_1 + r_0$ is 0; this means that there are no relevant documents and, therefore, p is undefined. The $\mathcal{A}'$ measure is defined for this scenario because it is a function of both p' and t', which are defined for this scenario.

Lemma 8.3.1. *The cardinality of the set of weak* 4*-compositions that correspond to Scenario 1 is* $N+1$.

Proof. The r_1 and r_0 parameters are effectively constants because their values are fixed at 0. The two remaining parameters can have any integer value in the closed interval $[0, N]$, subject to the constraint that $s_0 + s_1 = N$. Therefore, this problem has been reduced to finding the number of weak 2-compositions for N. By Equation 2.2.2 on page 26, this number is

$$\begin{aligned}\binom{N+2-1}{2-1} &= \binom{N+1}{1} \\ &= N+1.\end{aligned}$$

□

Scenario 2: None of the relevant documents contained the query term

The number of relevant documents with feature frequency 1 is fixed at 0 (i.e., $r_1 = 0$). The values of all the other parameters are free to vary.

Lemma 8.3.2. *The cardinality of the set of weak* 4*-compositions that correspond to Scenario 2 is* $\binom{N+2}{2}$.

Proof. The r_1 parameter is effectively a constant because its value is fixed at 0. The three remaining parameters can have any integer value in the closed interval $[0, N]$, subject to the constraint that $s_0 + r_0 + s_1 = N$. Therefore, this problem has been reduced to finding the number of weak 3-compositions for N. By Equation 2.2.2 on page 26, this number is

$$\binom{N+3-1}{3-1} = \binom{N+2}{2}.$$

□

Scenario 3: None of the documents contained the query term

The number of relevant and non-relevant documents with feature frequency 1 is fixed at 0 (i.e., $r_1 + s_1 = 0$). The values of all the other parameters are free to vary.

Lemma 8.3.3. *The cardinality of the set of weak* 4*-compositions that correspond to Scenario 3 is* $N + 1$.

Proof. The r_1 and s_1 parameters are effectively constants because their values are fixed at 0. The two remaining parameters can have any integer value in the closed interval $[0, N]$, subject to the constraint that $s_0 + r_0 = N$. Therefore, this problem has been reduced to finding the number of weak 2-compositions for N. By Equation 2.2.2 on page 26, this number is

$$\begin{aligned}\binom{N+2-1}{2-1} &= \binom{N+1}{1} \\ &= N+1.\end{aligned}$$

□

Scenario 4: All the relevant documents contained the query term

The number of relevant documents with feature frequency 1 is positive (i.e., $r_1 > 0$), the number of relevant documents with feature frequency 0 is zero (i.e., $r_0 = 0$). The values of all the other parameters are free to vary.

Lemma 8.3.4. *The cardinality of the set of weak* 4*-compositions that correspond to Scenario 4 is* $\binom{N+1}{2}$.

Proof. The r_0 parameter is effectively a constant because its value is fixed at 0. The r_1 parameter must have a value in the closed interval $[0, 1]$. The two remaining parameters can have any integer value in the closed interval $[0, N]$, subject to the constraint that

$r_1 + s_0 + s_1 = N$. Therefore, this problem has been reduced to finding the number of weak 3-compositions for $N - 1$. By Equation 2.2.2 on page 26, this number is

$$\binom{(N-1)+3-1}{3-1} = \binom{N+1}{2}.$$

□

Scenario 5: All documents contained the query term

The number of documents with feature frequency 1 is positive (i.e., $r_1 > 0, s_1 > 0$), the number of documents with feature frequency 0 is zero (i.e., $r_0 + s_0 = 0$).

Lemma 8.3.5. *The cardinality of the set of weak* 4*-compositions that correspond to Scenario 5 is* $N - 1$.

Proof. The s_0 and r_0 parameters are effectively constants because their values are fixed at 0. The two remaining parameters can have any integer value in the closed interval $[1, N]$, subject to the constraint that $r_1 + s_1 = N$. Therefore, this problem has been reduced to finding the number of (strong) 2-compositions for $N - 2$. By Equation 2.2.1 on page 26, this number is

$$\binom{N-1}{2-1} = \binom{N-1}{1}$$
$$= N - 1.$$

□

8.3.2 The Determination of Cardinalities for Two Combined Sets of Boundary Conditions

In this subsection, we determine the cardinalities of disjoint sets of boundary conditions. More specifically, we determine the cardinality of the combined sets of weak 4-compositions that correspond to the first three scenarios (i.e., Scenarios 1, 2, 3) and the last two scenarios (i.e., Scenarios 4, 5). These sets of scenarios corresponded, respectively, to (a) the total number of weak 4-compositions when none of the relevant documents contained the query term and (b) the number of weak 4-compositions when there was at least one relevant document and every relevant document contained the query term. The remainder of this subsection discusses how to determine the values for (a) and (b).

The Number of Weak 4-Compositions When None of the Relevant Documents Contain the Query Term

The goal of the next few paragraphs is to determine the cardinality of the combined sets of weak 4-compositions that correspond to the first three scenarios (i.e., Scenarios 1, 2, 3). A reasonable first attempt at determining this grand cardinality would simply sum the cardinalities for the first three scenarios. The sum of these cardinalities is

$$(N+1) + \binom{N+2}{2} + (N+1) = 2(N+1) + \binom{N+2}{2}.$$

Unfortunately, this sum counts some elements twice and others thrice. For example, when N equals 5, $(r_1, s_0, r_0, s_1) = (0, 4, 0, 1)$ is counted twice and $(r_1, s_0, r_0, s_1) = (0, 5, 0, 0)$ is counted three times. The reason for this overcount is indicated by the r_1, s_0, r_0, s_1 patterns in Table 8.5 on page 354.

These patterns show that some members of the set that corresponds to the situation where none of the documents are relevant (i.e., Scenario 1) are also members of the set that corresponds to the situation where none of the documents contain the query term (i.e., Scenario 3), and *vice versa*. Also, they show that some members of the set for the situation that corresponds to none of the relevant documents containing the query term (i.e., Scenario 2) are members of the other two sets. Finally, they show that all members of the sets for Scenarios 1 and 3 are members of the set for Scenario 2. Hence, the set of weak 4-compositions that are associated with the pattern for Scenario 2 subsumes the sets of weak 4-compositions that are associated with the patterns for Scenarios 1 and 2.

Lemma 8.3.6. *The cardinality of the total set of weak* 4*-compositions that correspond to the combined patterns for Scenarios 1, 2, and 3 is the same as the cardinality of the set of weak* 4*-compositions that correspond to the pattern for Scenario 2.*

Proof. The r_1, s_0, r_0, s_1 patterns in Table 8.5 on page 354 show that the set of weak 4-compositions for Scenario 2 subsumes those for Scenarios 1 and 3. Since the part of the patterns that are associated with the r_1 and s_0 parameters are identical for all three scenarios, we only need to inspect their r_0 and s_1 parts. The $\star$'s in the r_0 and s_1 parts for Scenario 2 are associated with either another $\star$ or a zero for Scenarios 1 and 3. Due to a $\star$ having the same meaning in each of the scenarios, and also to it being more general than a zero, the r_0 and s_1 parts for Scenario 2 subsume their counterparts for Scenarios 1 and 3. From this we can conclude that the set of weak 4-compositions for Scenario 2 subsume those for Scenarios 1 and 3. □

The Number of Weak 4-Compositions When There is at Least One Relevant Document and Every Relevant Document Contains the Query Term

Similar to the goal for the first three scenarios, the goal of the next few paragraphs is to determine the cardinality of the combined sets of weak 4-compositions that correspond to the last two scenarios (i.e., Scenarios 4,5). Like before, a reasonable first attempt at determining this grand cardinality would simply sum the cardinalities for the last two scenarios. The sum of these cardinalities is

$$N - 1 + \binom{N+1}{2}.$$

Unfortunately, this sum counts some elements twice. For example, when N equals 5, $(r_1, s_0, r_0, s_1) = (1, 0, 0, 4)$ is counted twice. The reason for this overcount is indicated by the r_1, s_0, r_0, s_1 patterns in Table 8.5 on page 354.

These patterns show that some members of the set that corresponds to the situation where every relevant document contains the query term (i.e., Scenario 4) are also members of the set that corresponds to the situation where every document contains the query term (i.e., Scenario 5) and all members of the set for Scenario 5 are members of the set for Scenario 4. Hence, the set of weak 4-compositions that are associated with the pattern for Scenario 5 subsumes the set of weak 4-compositions that are associated with the pattern for Scenario 5.

Lemma 8.3.7. *The cardinality of the total set of weak* 4*-compositions that correspond to the combined patterns for Scenarios 4 and 5 is the same as the cardinality of the set of weak* 4*-compositions that correspond to the pattern for Scenario 4.*

Proof. The r_1, s_0, r_0, s_1 patterns in Table 8.5 on page 354 show that the set of weak 4-compositions for Scenario 4 subsumes the one for Scenario 5. Since the part of the patterns that are associated with the r_1 and r_0 parameters are identical for the two scenarios, we only need to inspect their s_0 and s_1 parts. The $\star$'s in the s_0 and s_1 parts for Scenario 4 are associated with either another $\star$ or a $+$ for Scenario 5. Due to a $\star$ having the same meaning in each of the scenarios and also to it being more general than a $+$, the s_0 and s_1 parts for Scenario 4 subsume their counterparts for Scenario 5. From this we can conclude that the set of weak 4-compositions for Scenario 4 subsumes the set for

Scenario 5. □

The Number of Weak 4-Compositions For the Combined Sets of Boundary Conditions

Table 8.6 lists the combined sets of boundary conditions. From the information there, the rest of this discussion only needs to concern itself with combined sets A and B. These are exactly the ones that represent the only weak 4-compositions where corresponding values of $\mathcal{A}$ and $\mathcal{A}'$ differ in a document collection of size N. The remainder of the proof for this assertion is handled by Lemma 8.3.8.

Table 8.6: Combined Sets of Boundary Conditions for $\mathcal{A}$ and $\mathcal{A}'$ (After Subsumption).

Combined Sets of Boundary Conditions	r_1	s_0	r_0	s_1	# weak 4-comps
A (1, 2, 3)	0	$\star$	$\star$	$\star$	$\binom{N+2}{2}$
B (4, 5)	$+$	$\star$	0	$\star$	$\binom{N+1}{2}$

Lemma 8.3.8. *The number of corresponding $\mathcal{A}$ and $\mathcal{A}'$ values that are different for the weak 4-compositions that are in Scenarios A and B in a document collection of size N is*

$$2\binom{N}{2}. \tag{8.3.1}$$

Proof. First, we must establish that the sets that are associated with the two combined scenarios are disjoint. This is readily done by noticing that the r_1 parameter for the first combined scenario must always be a zero whereas the one for the second combined scenario must always be a positive number. Clearly, on this basis alone, the sets must be disjoint. The advantage to determining that these sets are disjoint is that the calculations for the number of situations in each set can be performed independently of each other. Once these numbers have been determined, we can simply add them in order to obtain the overall, or grand, total because we do not have to worry about the possibility of any overlap between the members of the sets that correspond to Combined Sets A and B.

Second, using the information in Tables 8.5 on page 354 and 8.6 on the previous page, we calculate the cardinalities for each combined scenario in the cases below and total them. The expression for Equation 8.3.1 on the preceding page is the simplified sum of the expressions in Equation 8.3.2 on the next page and Equation 8.3.3 on page 364 .

Combined Set A (the combining of boundary condition sets 1, 2, and 3).

Several members of the set of weak 4-compositions corresponding to this combined set of boundary conditions have an undefined p value when their r_1 and r_0 values are both 0. For each of these members, $\mathcal{A}$ does not have a value because p is undefined. $\mathcal{A}'$, by contrast, has a value of $1 - \epsilon$. Hence, $\mathcal{A}$ and $\mathcal{A}'$ are incomparable for these members, and there are

$$\binom{N+2-1}{2-1} = N + 1$$

of them because $s_1 + s_0 = N$ when $r_1 + r_0 = 0$. Basically, due to these conditions, the problem of determining how many members of this kind that are in the combined set can be reduced to the problem of determining the number of weak 2-compositions for N. This is indicated by the entries in the first row of Table 8.7 on the next page. The number of weak 4-compositions for the other two conditions that are enumerated in this table can be determined in a similar manner.

From the pattern for the $\mathcal{A} = \mathcal{A}'$ condition, the number of weak 4-compositions for this pattern reduces to determining the number of weak 2-compositions for $N - 1$ because $r_1 = s_1 = 0$ and the value of s_1 must be at least 1. Therefore, the number of weak 4-compositions for this pattern is

$$\binom{(N-1)+2-1}{2-1} = \binom{N}{1} = N.$$

The number of weak 4-compositions for the $\mathcal{A} \neq \mathcal{A}'$ condition reduces to determining the number of weak 3-compositions for $N - 2$ because $r_1 = 0$ and the values of r_0 and s_1 must be at least 1. Therefore, the number of weak 4-compositions that are associated with this condition is

$$\binom{(N-2)+3-1}{3-1} = \binom{N}{2}. \tag{8.3.2}$$

Table 8.7: Combined Set of Boundary Conditions A (The Number of Weak 4-Compositions When None of the Relevant Documents Contain the Query Term).

Condition	r_1	s_0	r_0	s_1	# weak 4-comps
$\mathcal{A}$ is undefined	0	$\star$	0	$\star$	$N+1$
$\mathcal{A} = \mathcal{A}'$	0	$\star$	$+$	0	N
$\mathcal{A} \neq \mathcal{A}'$	0	$\star$	$+$	$+$	$\binom{N}{2}$

These three conditions are mutually exclusive. This fact can be verified by showing that the sums of the expressions for their respective numbers of weak 4-compositions total $\binom{N+2}{2}$, the expression that appears in the first row of Table 8.6 on page 361, i.e.,

$$\begin{aligned}(N+1) + N + \binom{N}{2} &= (2N+1) + N(N-1)/2 \\ &= (4N + 2 + N(N-1))/2 \\ &= (4N + 2 + N^2 - N)/2 \\ &= (3N + 2 + N^2)/2 \\ &= (N+2)(N+1)/2 \\ &= \binom{N+2}{2}.\end{aligned}$$

Combined Set B (the combining of boundary condition sets 4 and 5).

From the pattern for the $\mathcal{A} = \mathcal{A}'$ condition, the number of weak 4-compositions for this pattern reduces to determining the number of weak 2-compositions for $N - 1$ because $s_0 = r_0 = 0$ and the value of r_1 must be at least 1. Therefore, the number of weak 4-compositions for this pattern is

$$\binom{(N-1)+2-1}{2-1} = \binom{N}{1} = N.$$

The number of weak 4-compositions for the $\mathcal{A} \neq \mathcal{A}'$ condition reduces to determining the number of weak 3-compositions for $N - 2$ because $r_0 = 0$ and the values of r_1 and s_0 must be at least 1. Therefore, the number of weak 4-compositions that are associated with this condition is

$$\binom{(N-2)+3-1}{3-1} = \binom{N}{2}. \tag{8.3.3}$$

Table 8.8: Combined Set of Boundary Conditions B (The Number of Weak 4-Compositions When There is at Least One Relevant Document and Every Relevant Document Contains the Query Term).

Condition	r_1	s_0	r_0	s_1	# weak 4-comps
$\mathcal{A} = \mathcal{A}'$	$+$	0	0	$\star$	N
$\mathcal{A} \neq \mathcal{A}'$	$+$	$+$	0	$\star$	$\binom{N}{2}$

These two conditions are mutually exclusive. This fact can be verified by showing that the sums of the expressions for their respective numbers of weak 4-compositions total $\binom{N+1}{2}$, the expression that appears in the second row of Table 8.6 on page 361, i.e.,

$$\begin{aligned} N + \binom{N}{2} &= N + N(N-1)/2 \\ &= (2N + N(N-1))/2 \\ &= (2N + N^2 - N)/2 \\ &= (N^2 + N)/2 \end{aligned}$$

$$= N(N+1)/2$$
$$= \binom{N+1}{2}.$$

□

8.3.3 Test Data Generation

The information in Figure 8.3 on page 353 and Figure 8.4 on page 353 was used to help construct two test programs. The first program generated all the weak 4-compositions for document collections where $1 \leq N \leq 200$. The second program only generated weak 4-compositions for 11 of the 12 categories listed in these figures. That is, it only generated compositions for the boundary conditions. The excluded category was the one in the former figure where the conditions $0 < p < 1$ and $0 < t < 1$ are jointly true. This latter program handled verification for collections where $201 \leq N \leq 400$. There were two main reasons for working with restricted versions of the sample space for this program: to establish more confidence in the formula for $\mathcal{A}'$ and because of the adverse effects of combinatorial explosion that were observed with the execution of the first test program.

Each weak 4-composition in the first test program was assigned to 1 of 12 mutually exclusive categories. Nine of the categories came from those listed in Figure 8.3 on page 8.3, the other three came from Figure 8.4 on page 353. The conditions in the box for each category in the figures specify the validation criteria for that category.

For example, when the joint condition $p = 0$ and $t = 0$ holds for a given weak 4-composition, it should also be the case that both the corresponding p' and t' values are equal to the appropriate ϵ for the number of documents in that collection. The test program computed these values and checked to see if they satisfied the expected conditions. The test program also checked to see if the computed $\mathcal{A}$ and $\mathcal{A}'$ values were equal. If an affirmative answer was obtained for both of these situations, then the given weak 4-composition passed validation; if not, then it failed validation.

The generation and validation of weak 4-compositions, for even small to moderate values of N, was time- and memory-intensive. For example, it required over 18 hours of real-time on the writer's personal computer to generate and validate all of the weak 4-compositions for $1 \leq N \leq 200$. The major reason for this was that a large number of sample points had to be generated due to the cardinality of the set of weak 4-compositions for N being a cubic function of N. This cardinality grew very rapidly as N increased. The number of weak 4-compositions that had to be generated and validated for just the first 200 positive values of N was

$$\sum_{N=1}^{200} \binom{N+3}{3} = 70,058,750.$$

As was mentioned earlier, the second test program validated weak 4-compositions when $201 \leq N \leq 400$. The problem of combinatorial explosion, even for values of N as small as a few hundred, was daunting during this validation process and became exponentially more so as the value of N increased. The time involved in all the weak 4-compositions for $201 \leq N \leq 400$ would take several days on the writer's computer. Since (1) the largest number of weak 4-compositions fell into the category where the conditions $0 < p < 1$ and $0 < t < 1$ were jointly true and (2) the results obtained by the first test program passed validation, it was decided to exclude, from validation, the weak 4-compositions from (1). This left, for validation, only the weak 4-compositions that were associated with the boundary conditions.

By the information in Table 8.6 on page 361, the second test program only had to validate

$$\binom{N+2}{2} + \binom{N+1}{2}$$

weak 4-compositions for each value of N. The growth rate of the cardinality of this reduced set of weak 4-compositions was quadratic (shown by Lemma 8.3.9), rather than cubic as was the growth rate for the non-reduced set of weak 4-compositions. In practical terms, this made the additional validation efforts feasible for $201 \leq N \leq 400$ because using an unrestricted sample space would result in the generation and validation of slightly over one billion weak 4-compositions (demonstrated by the sum for Equation 8.3.4) versus approximately 19 million of these kinds of weak compositions (demonstrated by the sum for Equation 8.3.5 on page 367). The difference between these two sums was over an order of magnitude.

$$\sum_{N=201}^{400} \binom{N+3}{3} = 1,023,508,750. \tag{8.3.4}$$

$$\sum_{N=201}^{400} \left(\binom{N+2}{2} + \binom{N+1}{2} \right) = 18,847,100. \tag{8.3.5}$$

Lemma 8.3.9. *The growth rate of the cardinality of the reduced set of weak* 4*-compositions for a document collection of size N is quadratic.*

Proof.

$$\begin{aligned}
\binom{N+2}{2} + \binom{N+1}{2} &= \frac{(N+2)(N+1)}{2} + \frac{(N+1)N}{2} \\
&= \frac{N^2 + 3N + 2 + N^2 + N}{2} \\
&= \frac{2N^2 + 4N + 2}{2} \\
&= N^2 + 2N + 1 \\
&= \Theta(N^2).
\end{aligned}$$

□

8.3.4 Empirical Data Supports the Validation of $\mathcal{A}'$

Two Mathematica® (Wolfram, 2003) programs were written to implement the test programs. One program performed exhaustive testing for the sets of weak 4-compositions for document collections where $1 \leq N \leq 200$ and the other program did boundary condition testing for document collections where $201 \leq N \leq 400$.

These test programs were run and the expected values were compared to the actual values. The expected and actual values matched exactly for all 70,058,750 weak 4-compositions generated and examined by the first test program. The same results were observed for the 18,847,100 weak 4-compositions examined by the second test program. In addition to performing 11 of the 12 category tests that the program for exhaustive testing did, the second program also computed the numbers of weak 4-compositions that met the three conditions in Table 8.7 on page 363 and the 2 conditions in Table 8.8 on page 364. These expected number were compared to the actual numbers of weak 4-compositions for the sets that corresponded to each of these 5 conditions. In all cases, the expected and actual values were identical. For example, when $N = 250$, the expected values were, respectively,

$$\begin{aligned}
\text{card}(\mathcal{A} \text{ is undefined, combined set A, 250}) &= 50 + 1 = 251, \\
\text{card}(\mathcal{A} = \mathcal{A}', \text{combined set A, 250}) &= 250, \\
\text{card}(\mathcal{A} \neq \mathcal{A}', \text{combined set A, 250}) &= \binom{250}{2} = 31,125, \\
\text{card}(\mathcal{A} = \mathcal{A}', \text{combined set B, 250}) &= 250, \text{ and} \\
\text{card}(\mathcal{A} \neq \mathcal{A}', \text{combined set B, 250}) &= \binom{250}{2} = 31,125,
\end{aligned}$$

where card(*cond, combSet*, N) denotes the cardinality of the combined set *combSet*, for a document collection of size N, when its members are restricted to those that satisfy condition *cond*. The conclusion from the results of these tests was an extremely high confidence level that the equation developed in this dissertation for calculating the $\mathcal{A}'$ measure is correct.

8.4 The Validation of ASL'

The vast majority of the effort that was involved in validating ASL' was subsumed by the validation efforts for $\mathcal{Q}'$ and $\mathcal{A}'$. The major remaining tasks were to compare selected ASL values with selected ASL' values for several document collection sizes.

The weak 4-compositions (that the ASL' values are based on) for a specific N were those with r_1 and r_0 components that indicated there was at least one relevant document (i.e., $r_1 + r_0 > 0$ was true) for the associated query. This resultant set of weak 4-compositions has almost as many members as an unfiltered set because the $r_1 + r_0 > 0$ filter only excludes $N + 1$ members of the unfiltered set. This is a negligible amount of members to exclude because

$$\lim_{N\to\infty} \frac{N+1}{\binom{N+3}{3}} = 0.$$

This resultant set for N was divided into two groups. The first group contained all the members for which the conditions $0 < p < 1$ and $0 < t < 1$ were simultaneously true; the second group contained only members not meeting either of those conditions. The members in the first group were those that should have identical ASL and ASL' values because their corresponding p and p' values should be identical and their corresponding t and t' values should also be identical. The members of the second group were all those where the singularity-handling technique developed in a previous chapter might have an impact on the calculation of the p' and t' values. In general, the members of this group were normally expected to have different p and p' values. Similarly, they were also expected to normally have different t and t' values.

8.4.1 Test Data Generation

The set W of weak 4-compositions for a document collection of size 200 was created. Three mutually exclusive subsets A, B, and C were created from W in the following way: the 201 members of W that corresponded to weak 4-compositions that had no relevant documents were placed in A, all the members of W where the conditions $0 < p < 1$ and $0 < t < 1$ were simultaneously true were placed in B, and the remaining members of W were placed in C.

8.4.2 Empirical Data Supports the Derivation of ASL'

Since the ASL is undefined for queries that do not have any relevant documents, the part of the validation process that involved members of set A only needed to compute the ASL' values for the 201 members of set A and compare them to the manually-calculated ASL' values for these members. This was done and it was verified that the actual values and the expected values were exact matches.

The ASL value for each member of set B was expected to be equal to its ASL' counterpart. This expectation was verified for each member of B.

The ASL and ASL' values were computed for each member of C and compared to each other. Except for the situation where

$$p - t = p' - t',$$

for a member of C, the ASL and its ASL' values are expected to be different. This was verified to be true for each member of C.

8.5 The Validation of $\mathrm{ASL}'_{\mathrm{r}}$

The validation of this variant of the ASL' measure, namely, $\mathrm{ASL}'_{\mathrm{r}}$, first had to develop criteria for the validation.The first set of criteria were conditions for which the $\mathrm{ASL}'_{\mathrm{r}}$, and ASL' values should always agree for a specific weak 4-composition and ranking method. Two conditions emanated from this phase. The second set of criteria involved the handling of expected disagreements between these two values. Conditions were involved to determine when the $\mathrm{ASL}'_{\mathrm{r}}$ value was “better” than the ASL' value. *Better* was defined to mean that the absolute difference between the $\mathrm{ASL}'_{\mathrm{r}}$ value and the value v calculated by an actual ranking was less than or equal to the absolute difference between the ASL' value and v.

The general condition came directly from the top part of Equation 7.10.2 on page 331:

$$(\mathcal{Q}' > 0 \text{ and } \mathrm{dtw}_{\mathrm{rm}} \geq 0) \text{ or } (\mathcal{Q}' = 0 \text{ and } \mathrm{dtw}_{\mathrm{rm}} < 0). \tag{8.5.1}$$

The second condition was obtained by finding all the solutions of

$$f_{\mathrm{ASL}}(N, \mathcal{Q}', \mathcal{A}') - f_{\mathrm{ASL}}(N, \mathcal{Q}', 1 - \mathcal{A}') = 0.$$

After expanding the f_{ASL} references in this equation, we obtained

$$\mathcal{Q}'\mathcal{A}' + (1 - \mathcal{Q}')(1 - \mathcal{A}') - (\mathcal{Q}'(1 - \mathcal{A}') + (1 - \mathcal{Q}')\mathcal{A}') = 0.$$

Simplification yielded

$$4\mathcal{Q}'\mathcal{A}' - 2\mathcal{Q}' - 2\mathcal{A}' + 1 = 0.$$

Factoring produced

$$(2\mathcal{Q}' - 1)(2\mathcal{A}' - 1) = 0.$$

Visual inspection indicated that the set of solutions for this equation was

$$\{\mathcal{Q}' = 1/2, \mathcal{A}' = 1/2\},$$

meaning that the second condition was

$$\mathcal{Q}' = 1/2 \text{ or } \mathcal{A}' = 1/2. \tag{8.5.2}$$

The third condition was used to determine compare if the absolute difference between ASL'_r and ASL'_g was no greater than the absolute difference between ASL' and ASL'_g. It was defined as

$$|\mathrm{ASL}'_\mathrm{r} - \mathrm{ASL}'_\mathrm{g}| \leq |\mathrm{ASL}' - \mathrm{ASL}'_\mathrm{g}|. \tag{8.5.3}$$

8.5.1 Test Data Generation

Five sets of test queries were constructed to help determine if the values computed by the ASL'_r formula were correct. The query sets were for document collection sizes of 10, 15, 20, 25, and 30. The set of test queries for a collection of size N have a one-to-one correspondence with those in the set of weak 4-compositions of N, after the weak compositions that correspond to zero relevant documents have been excluded.

Five datasets were created, one for each set of test queries. Each test dataset has 8 columns:

query, ranking_method,
ASL', ASL'Refined, ASL'Gold,
condition1, condition2, and condition3.

The first column (query) identified the query (represented as a weak 4-composition). The second column (ranking_method) identified the ranking method. The third through fifth columns represented the calculated values of the ASL' measure and its ASL'_r and ASL'_g variants. The condition1, condition2, and condition3 columns represented the Boolean values of true and false for Condition 8.5.1 on page 371, Condition 8.5.2 on the previous page, and Condition 8.5.3 on the preceding page, respectively. Each test dataset had

$$\sum_{N\in\{10,15,20,25,30\}} \left(\binom{N+3}{3} - (N+1)\right) = 11,605 \tag{8.5.4}$$

rows. Collectively, the five datasets had a total of $5 \times 11605 = 86,025$ rows.

8.5.2 Empirical Data Supports the Validation of ASL'_r

The values in each dataset were checked to verify that the calculated ASL'_r value for a query and ranking method combination was always at least as good as the corresponding ASL' value for that same combination. The other major check for a query and ranking method combination was to ensure that its values were identical if and only if either one, or both, of Condition 8.5.1 on page 371 and Condition 8.5.2 on the preceding page were true.

These conditions were found to be always true for each of the query and ranking method combinations in the 5 datasets. From these results, the conclusion was that the formula for the ASL'_r variant calculated the expected results.

8.6 The Validation of $\mathrm{ASL}'_{\mathrm{g}}$

This particular variant of the ASL' measure, namely, $\mathrm{ASL}'_{\mathrm{g}}$, was introduced in Section 7.10.2, starting on page 331. Its value is considered to be the gold standard value for the ASL' measure. It is the same value that would be obtained by the following empirical process: generate all the possible document sequences for a particular ranking method rm, document collection c, and query q; calculate each sequence's MSL value; and, finally, compute the arithmetic mean of these MSL values. The resultant value is the $\mathrm{ASL}'_{\mathrm{g}}$ value.

The main idea behind using analytic techniques to help calculate the $\mathrm{ASL}'_{\mathrm{g}}$ measure, as contrasted to the empirical process that was described in the previous paragraph, is, ideally, to be able to calculate this measure from just the values of certain parameters (e.g., r_1, r_0, s_1, s_0), that we are already familiar with from previous chapters, and the feature weight equations, that are given in Table 8.9 on page 376, without having to physically rank the documents and then having to use brute force techniques to enumerate all the possible document sequences. Three methods were developed to check the correctness of this measure. By the way, this is another example of the strategy of triangulation (i.e., using multiple techniques to ascertain the correctness of this new measure).

For the convenience of the reader, the information in Chapter 7 from Table 7.1 on page 329, Figure 7.9 on page 329, and Table 7.2 on page 333, respectively, is repeated in Table 8.9 on page 376, Figure 8.1 on page 376, and Table 8.10 on page 376. Also, for the convenience of the reader, we review the notation that is used in Table 8.10 on page 376. The $\mathrm{dtw}_{\mathrm{rm}}$ term denotes the document term weight for ranking method rm (i.e., best-case, coordination level matching, decision-theoretic, inverse document frequency, random, worst-case); n_{R} denotes the number of documents at the rear of a ranking; n_{RR} denotes the number of relevant documents that are among the n_{R} documents at the rear of a ranking; n_{F} denotes the number of documents at the front of a ranking; and n_{RF} denotes the number of relevant documents that are among the n_{R} documents at the front of a ranking.

Three Methods That Calculate the $\mathrm{ASL}'_{\mathrm{g}}$ Measure

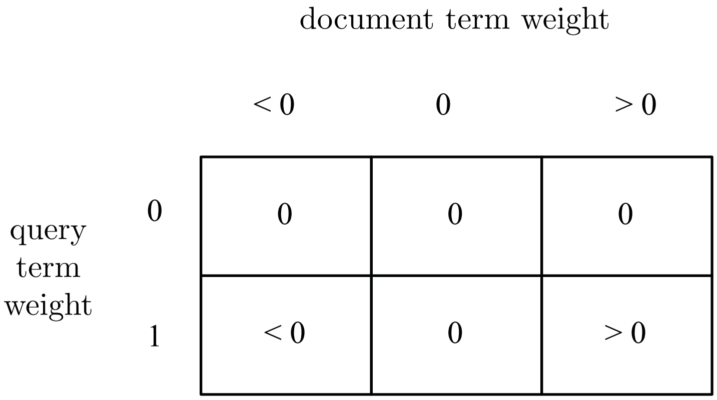
Method H is a hybrid method because it combines empirical and analytic techniques. It computes the ASL value by first non-ascendingly sorting the documents by their RSVs. For a given query q, document collection c, and ranking method rm, the sorting partitions the documents into, at most, two clusters. The number of documents in each partition, along with the number of relevant documents in each partition, and the relative positions of these partitions are used as parameters to the probability generating functions (PGFs) to calculate the value of the $\mathrm{ASL}'_{\mathrm{g}}$ measure.

Method P uses the information in Table 8.9 on the following page and in Table 8.10 on page 376, in conjunction with a query q; a ranking method rm; the document term weight function for this method; and the values of $r_1, r_0, s_1,$ and s_0 to set up a mathematical model that uses PGFs to calculate $\mathrm{ASL}'_{\mathrm{g}}$. The documents do not need to be sorted and are not sorted. Based on the parameter values for r_1, r_0, s_1, and s_0, and the weighting function for a particular ranking method *rm,* we can determine if the document term weight is negative, zero, or positive. Once we know this, we can use the information in Table 8.9 on the next page and Table 8.10 on the following page to help construct a PGF for the $\mathrm{ASL}'_{\mathrm{g}}$ measure.

Method C also uses the information in Table 8.9 on the next page and Table 8.10 on the following page. Unlike Method P, which uses this information to construct a PGF for the $\mathrm{ASL}'_{\mathrm{g}}$ measure, this method uses this information to develop a set of combinatoric-based equations to calculate the $\mathrm{ASL}'_{\mathrm{g}}$ measure.

Table 8.9: Feature Weights for Several Ranking Methods.

Ranking Method	Feature Weight
Best-case	$w = \log\left(\frac{p/(1-p)}{t/(1-t)}\right)$
Worst-case	$w = -\log\left(\frac{p/(1-p)}{t/(1-t)}\right)$
Random	$w = \begin{cases} \text{Best-case weight} & : \ 1/2 \text{ of the time;} \\ \text{Worst-case weight} & : \ 1/2 \text{ of the time.} \end{cases}$
Decision-theoretic	$w = \log\left(\frac{p/(1-p)}{q/(1-q)}\right)$
Inverse document frequency	$w = -\log(t)$
Coordination-level matching	$w = c$ (a positive constant)



document term weight

query term weight		<0	0	>0
	0	0	0	0
	1	<0	0	>0

Figure 8.1: RSVs and Their Relation to Query and Document Weights.

Table 8.10: Document Distribution at the Front and Rear of An Actual Ranking.

condition	n_{R}	n_{RR}	n_{F}	n_{RF}
$\mathrm{dtw}_{\mathrm{rm}} > 0$	n_0	r_0	n_1	r_1
$\mathrm{dtw}_{\mathrm{rm}} = 0$	$n_1 + n_0$	$r_1 + r_0$	0	0
$\mathrm{dtw}_{\mathrm{rm}} = 0$	0	0	$n_1 + n_0$	$r_1 + r_0$
$\mathrm{dtw}_{\mathrm{rm}} < 0$	n_1	r_1	n_0	r_0

8.6.1 Test Data Generation

Five sets of test queries were constructed to help determine if the ASL'_g formulas developed by the three methods just described were in total agreement on the values that they calculated. The query sets were for document collection sizes of 10, 15, 20, 25, and 30. The set of test queries for a collection of size N have a one-to-one correspondence with those in the set of weak 4-compositions of N, after the weak compositions that correspond to zero relevant documents have been excluded.

Five datasets were created, one for each set of test queries. All ranking methods were tested, except for random ranking. The random ranking method was excluded because, for a particular query, the actual ranking method that it uses is always going to be either the best case or worst case ranking method. It is assumed that if these latter two ranking methods pass validation, then so would the random ranking method because it it based on these two methods. The probability is 0.5 that best case ranking is arbitrarily chosen to implement random ranking; likewise, the probability is 0.5 that the method chosen would be worst case ranking.

Each test dataset had 4 columns:

query,
ASL'Gold_mH, ASL'Gold_mP, and ASL'Gold_mC.

The first column (query) was the weak 4-composition that represents the query and the remaining columns represented the ASL'_g values for this query that were computed by Methods H, P, and C, respectively. Each test dataset has

$$\sum_{N\in\{10,15,20,25,30\}} \left(\binom{N+3}{3} - (N+1)\right) = 11,605$$

rows. Collectively, the five datasets had a total of $5 \times 11605 = 86,025$ rows where all three ASL'_g values in a row must be equal to each other.

8.6.2 Empirical Data Supports the Validation of $\mathrm{ASL}'_{\mathrm{g}}$

Software was written to compare all three $\mathrm{ASL}'_{\mathrm{g}}$ values in each row of a test dataset. If the values were found to be identical, an initially zero-valued counter for that test dataset was incremented by 1. After all the rows for this dataset had been processed, the counter value was compared to 11,605. If these values were equal to each other, the validation of the three methods for that dataset (and associated ranking method) was considered a success.

This testing and validation procedure was followed for all 5 test datasets. In each row of these datasets, the three $\mathrm{ASL}'_{\mathrm{g}}$ values were found to be equal to each other. The conclusion was that the either of the formulas for these three methods could be used to correctly determine the exact $\mathrm{ASL}'_{\mathrm{g}}$ value.

8.6.3 An Example That Illustrates the Calculation of $\mathrm{ASL}'_{\mathrm{g}}$ By Three Different Methods

The data that appears in Figure 7.2 on page 266 is used in this example to calculate the value of the $\mathrm{ASL}'_{\mathrm{g}}$ measure for the weak 4-composition $(r_1, r_0, s_1, s_0) = (2, 1, 1, 4)$. The documents that correspond to this composition are ordered by the decision-theoretic (DT) ranking method.

The Hybrid Method (Method H)

The first step of this method is to calculate the retrieval status value (RSV) for each document. Table 8.9 on page 376 and Table 8.10 on page 376 contain information that indicates the RSV is 0 for any document that does not contain the query term and it is

$$\mathrm{RSV} = \log\left(\frac{p/(1-p)}{q/(1-q)}\right)$$

for any document that contains the query term. In this example,

$$\begin{aligned} p &= r_1/(r_1 + r_0) \\ &= 2/3 \end{aligned}$$

and

$$\begin{aligned} q &= s_1/(s_1 + s_0) \\ &= 1/5. \end{aligned}$$

Therefore, the retrieval status value for any document that contains the query term is

$$\begin{aligned} \mathrm{RSV} &= \log\left(\frac{(2/3)/(1-2/3)}{(1/5)/(1-1/5)}\right) \\ &= \log\left(\frac{(2/3)/(1/3)}{(1/5)/(4/5)}\right) \\ &= \log\left(\frac{2}{1/4}\right) \\ &= \log(8) \\ &= 2.07944. \end{aligned}$$

After assigning each document an RSV, the documents are non-ascendingly sorted by their RSVs. This results in three documents at the front of the ranked vector V of documents that have 2.07944 as their RSV, followed by five documents that have an RSV of 0. Two of the three front documents are relevant and only one of the documents at positions 4-8, inclusive, in V are relevant. We can use this information to construct the probability generating function (PGF) for the $\mathrm{ASL}'_{\mathrm{g}}$ measure with respect to this situation. Equation 7.5.9 on page 306 describes its probability generating function. This

PGF can be stated as

$$p(x) = (x^{7/3} + 2x^{8/3} + 3x^{9/3} + 3x^{10/3} + 3x^{11/3} + 2x^{12/3} + x^{13/3})/15.$$

If we take the first derivative of this function, with respect to x, we obtain

$$p'(x) = \frac{1}{15}\left(\frac{7x^{4/3}}{3} + \frac{16x^{5/3}}{3} + 9x^2 + 10x^{7/3} + 11x^{8/3} + 8x^3 + \frac{13x^{10/3}}{3}\right).$$

Finally, by setting x to 1, we obtain

$$\begin{aligned}\mathrm{ASL}'_{\mathrm{g}} &= p'(1)\\ &= (7/3 + 16/3 + 9 + 10 + 11 + 8 + 13/3)/15\\ &= 50/15\\ &= 10/3.\end{aligned}$$

The Probability Generating Function Method (Method P)

Unlike Method H, this method uses solely analytical means to obtain the value of $\mathrm{ASL}'_{\mathrm{g}}$. The parameter values that are used to construct the probability generating function are obtained from the information in Table 8.9 on page 376 and Table 8.10 on page 376.

Just as with Method H, we must first determine whether the RSV for a document that contains the query term is negative, zero, or positive. We proceed as in Method H and determine that the values of p, q, and the RSV are, respectively, 2/3, 1/5, and 2.07944. Since $(r_1, r_0, s_1, s_0) = (2, 1, 1, 4)$, we have $n_1 = r_1 + s_1 = 2 + 1 = 3$ and $n_0 = r_0 + s_0 = 1 + 4 = 5$. From this, and the information in Table 7.2 on page 333, we can state that

$$n_{\mathrm{R}} = n_0 = 5,$$

$$
\begin{aligned}
n_{\mathrm{RR}} &= r_0 = 1, \\
n_{\mathrm{F}} &= n_1 = 3, \text{ and} \\
n_{\mathrm{RF}} &= r_1 = 2.
\end{aligned}
$$

We can use this information to construct a probability generating function for $\mathrm{ASL}'_{\mathrm{g}}$. The ordinary generating function for the ranked documents that are at the front of the sequence is

$$
\begin{aligned}
\mathrm{FFfront}(x, z, n_{\mathrm{F}}) &= \prod_{i=1}^{n_{\mathrm{F}}} (1 + x^i z) \\
&= \prod_{i=1}^{3} (1 + x^i z).
\end{aligned}
$$

The analogous ordinary generating function for the ranked documents that are at the rear of the sequence is

$$
\begin{aligned}
\mathrm{FFrear}(x, y, n_{\mathrm{F}}, N) &= \prod_{i=n_{\mathrm{F}}+1}^{N} (1 + x^i y) \\
&= \prod_{i=3+1}^{8} (1 + x^i y) \\
&= \prod_{i=4}^{8} (1 + x^i y).
\end{aligned}
$$

The ordinary generating function for the entire ordering, $G_2(x, y, z)$, is the convolution of the ordinary generating functions for the two parts of the ordering, namely, $\mathrm{FFfront}(x, z, , n_{\mathrm{F}})$ and $\mathrm{FFrear}(x, y, n_{\mathrm{F}}, N)$:

$$
\begin{aligned}
G_2(x, y, z, n_{\mathrm{F}}, N) &= \mathrm{FFfront}(x, z, n_{\mathrm{F}}) \cdot \mathrm{FFrear}(x, y, n_{\mathrm{F}}, N) \\
&= \mathrm{FFfront}(x, z, 3) \cdot \mathrm{FFrear}(x, y, 3, 8).
\end{aligned}
$$

This equation is equivalent to Equation 7.5.7 on page 304. Let

$$\begin{aligned} F(x) &= [y^{n_{\mathrm{RR}}} z^{n_{\mathrm{RF}}}] \left(G_2(x, y, z, n_{\mathrm{F}}, N)|_{y=1,z=1} \right) \\ &= [y^1 z^2] \left(G_2(x, y, z, n_{\mathrm{F}}, N)|_{y=1,z=1} \right) \\ &= x^7 + 2x^8 + 3x^9 + 3x^{10} + 3x^{11} + 2x^{12} + x^{13}. \end{aligned}$$

be the expression that is obtained from the expansion of G_2 when the value 1 is substituted everywhere that a y or z appears in the expanded form. This resultant expression, denoted by $F(x)$, is now a function of just one variable, namely, x, because, for a given query, the values of $n_{\mathrm{F}}, n_{\mathrm{RF}}, n_{\mathrm{R}}, n_{\mathrm{RR}}$, and N can be treated as constants.

Next, we must adjust $F(x)$ for the number of relevant documents (i.e., $n_{\mathrm{RR}} + n_{\mathrm{RF}} = 1 + 2 = 3$) that is in each sequence. The result of this adjustment is

$$M(x) = x^{7/3} + 2x^{8/3} + 3x^{9/3} + 3x^{10/3} + 3x^{11/3} + 2x^{12/3} + x^{13/3}.$$

From this point onward, the rest of this example proceeds similarly to the example for Method H. That is,

$$\begin{aligned} p(x) &= M(x)/15 \\ &= (x^{7/3} + 2x^{8/3} + 3x^{9/3} + 3x^{10/3} + 3x^{11/3} + 2x^{12/3} + x^{13/3})/15 \end{aligned}$$

and

$$p'(x) = \frac{1}{15} \left(\frac{7x^{4/3}}{3} + \frac{16x^{5/3}}{3} + 9x^2 + 10x^{7/3} + 11x^{8/3} + 8x^3 + \frac{13x^{10/3}}{3} \right).$$

Finally, by evaluating the PGF $p'(x)$ at $x = 1$, we obtain

$$\begin{aligned}
\mathrm{ASL}'_{\mathrm{g}} &= p'(1) \\
&= (7/3 + 16/3 + 9 + 10 + 11 + 8 + 13/3)/15 \\
&= 50/15 \\
&= 10/3.
\end{aligned}$$

The Combinatoric Method (Method C)

This method is based on the result that was established by Lemma 7.10.1 on page 338, the proof of which relied on combinatoric arguments. According to that result,

$$\mathrm{ASL}'_{\mathrm{g}} = \frac{\mathrm{MSL}_{\mathrm{gold,front}} + \mathrm{MSL}_{\mathrm{gold,rear}}}{n_{\mathrm{RF}} + n_{\mathrm{RR}}}$$

where

$$\begin{aligned}
\mathrm{MSL}_{\mathrm{gold,front}} &= \frac{n_{\mathrm{RF}}(n_{\mathrm{F}} + 1)}{2}, \\
\mathrm{MSL}_{\mathrm{gold,rear}} &= \frac{n_{\mathrm{RR}}(\mathrm{n}_{\mathrm{R}} + 1)}{2} + \text{shift\_contribution}, \text{ and} \\
\text{shift\_contribution} &= n_{\mathrm{RR}} \cdot n_{\mathrm{F}}.
\end{aligned}$$

The previous discussion for Method P established that

$$\begin{aligned}
n_{\mathrm{R}} &= n_0 = 5, \\
n_{\mathrm{RR}} &= r_0 = 1, \\
n_{\mathrm{F}} &= n_1 = 3, \text{ and} \\
n_{\mathrm{RF}} &= r_1 = 2.
\end{aligned}$$

Based on that, we have

$$\begin{aligned}
\mathrm{MSL}_{\mathrm{gold,front}} &= \frac{2(3+1)}{2} = 4, \\
\mathrm{MSL}_{\mathrm{gold,rear}} &= \frac{1(5+1)}{2} + 1 \cdot 3 \\
&= 6, \text{ and} \\
\mathrm{ASL}'_{\mathrm{g}} &= \frac{4+6}{2+1} \\
&= 10/3.
\end{aligned}$$

8.7 Summary

This chapter discussed the validation efforts for the Average Search Length variants, the ranking method-specific quality of ranking measures, and the unnormalized average search length. The formulas for these entities were based on the discussions and work that were discussed in Chapters 4, 5, 6, and 7. More specifically, test data generation and the analysis of the studies that were performed on these test datasets were discussed for each of these entities: $\mathcal{Q}'$ estimates for the coordination level matching, inverse document frequency, and decision-theoretic ranking methods; the $\mathcal{Q}'$ estimates by random sampling; the unnormalized average search length $\mathcal{A}$; and the ASL', $\mathrm{ASL}'_{\mathrm{r}}$, and $\mathrm{ASL}'_{\mathrm{g}}$ measures. Each of these entities had a separate section in this chapter that was devoted to its test data dataset(s) and the analysis of the results that were obtained from performing various studies on the test data.

Chapter 9

The ASL Performance Measure Variants and Empirical Document Rankings

This chapter addresses the second of the three research questions that were enumerated in Section 3.5, which starts on page 103: Do the measures (i.e., ASL', $\mathrm{ASL}'_{\mathrm{r}}$) that estimate the ASL produce the same performance results as the measure (i.e., $\mathrm{ASL}'_{\mathrm{g}}$) that calculates the same results that would be produced by a process that ranks documents and, then, calculates the Average Search Length from this empirical ranking data?

This $\mathrm{ASL}'_{\mathrm{g}}$ measure calculates the same Average Search Length that would be calculated by empirical means, if the following actions were performed in this sequence: generate all the possible sequences of ranked documents for a query q, compute their respective ASL values, and then compute the mean of these values. The resultant value would be identical to the value for $\mathrm{ASL}'_{\mathrm{g}}$ (which is calculated by analytical means).

Section 3.5.2, which starts on page 106, contains the initial introduction for this research question. Much much information about the ASL', $\mathrm{ASL}'_{\mathrm{r}}$, and $\mathrm{ASL}'_{\mathrm{g}}$ measures can be found in Section 7.10, which starts on page 327. The ASL' measure is defined by Equation 7.10.1 on page 330, the $\mathrm{ASL}'_{\mathrm{r}}$ measure is defined by Equation 7.10.2 on page 331, the probability generating function version of the ASL'_g measure is defined by Equation 7.10.3 on page 335, and the combinatoric version of this measure is defined by Lemma 7.10.1 on page 336.

We start to answer this second research question by first establishing the appropriate hypotheses:

H_0: the estimated and empirical ASL measures for a ranking method produce the same results

H_1: the estimated and empirical ASL measures for a ranking method do not produce the same results.

The purpose of this chapter is to determine if three measures that calculate the ASL by different means are significantly different from each other at either the .05 or .01 significance levels. The ASL' and ASL'_r measures estimate the ASL by different means whereas the ASL'_g calculates, by analytic techniques, the same ASL value that would be obtained empirically if one used brute-force techniques to generate all the possible sequences for a query, calculated the ASL value for each sequence, and then computed the mean of these ASL values. The major difference between the ASL' and ASL'_r measures is that the ASL'_r measure incorporates two additional pieces of information: whether the value of the quality of ranking part of its formula is negative, zero, or positive and whether the document term weight part indicates a negative, zero, or positive value. In theory, if this additional information is known and and is incorporated in the estimation equation for the ASL, the ASL'_r measure should produce an estimate that is at least as close to the ASL'_g value as the ASL' value is.

The tests in this chapter compare the ASL' measure with the ASL'_g measure and the ASL'_r measure with the ASL'_g measure. In both of these comparisons, the ASL'_g measure is considered the *actual* ASL measure whereas the ASL' and ASL'_r measures are considered the *estimated* measures. In addition to these comparisons, the $\mathrm{ASL}'_{\mathrm{r}}$ and $\mathrm{ASL}'_{\mathrm{g}}$ measures are compared with each other. These tests were performed for each of the 6 ranking methods (i.e., best case, worst case, random, coordination level matching, inverse document frequency, and decision-theoretic).

The Kolmogorov-Smirnov test is the statistical hypotheses test that was used to assess the performance results. The Wilcoxon signed ranks test (with continuity correction) was used to help corroborate the results. The continuity correction version of the Wilcoxon signed ranks test was used due to the presence of many ties among the retrieval status values (RSVs) of the ranked documents. One of the primary reasons that these particular nonparametric tests were chosen, rather than a parametric one such as the paired t-test, is because the sampling distributions of several of the performance measures are unknown.

We were interested in testing at two significance levels: $\alpha = .05$ and $\alpha = .01$. The rejection region is two-tailed. The decision criterion is as follows. If the test statistic value falls in the rejection region, we can conclude that there was no statistically significant evidence of a difference in performance between the estimated and actual ASL' measures for a ranking method.

9.1 The Datasets

The queries and documents of the $\mathrm{CF}'_{\mathrm{combined}}$ collection were not used in the hypothesis testing because the data to the testing procedures was assumed to be random. An analysis of the queries showed that they did not have the characteristics that randomly-generated generated queries would have for this collection. This is corroborated by the information in Figure 3.4, on page 89, which shows that the randomly-generated synthetic test collection has very different performance characteristics, across all 6 of the ranking methods, than the parts of the graphs that corresponded to variants of the Cystic Fibrosis test collection.

In lieu of using the $\mathrm{CF}'_{\mathrm{combined}}$ collection, this study generated random queries for synthetic datasets of various sizes. In particular, the dataset sizes were 10 million, 100 million, one billion, and 10 billion documents. The numbers of random queries that were generated for each dataset are 100; 1,000; and 10,000. The results of using the Kolmogorov-Smirnov to analyze the performance measure data for these $4 \times 3 = 12$ combinations of dataset sizes and numbers of queries were used to construct 12 tables.

9.2 The Analysis

The performance measure data that was analyzed had many instances of ties (i.e., duplicate values). Due to the presence of these values, both the Kolmogorov-Smirnov and Wilcoxon signed ranks tests issued warnings about the ties. Due to the presence of these ties, neither test was able to calculate exact p-values; instead, approximate p-values were issued by these tests.

The inspection of the result tables showed that, across all 12 of the collection size and number of query combinations, for each ranking method, two of the comparisons exhibited statistically significant differences at both the $\alpha = .05$ and the $\alpha = .01$ significance levels. The remaining comparison showed that there was no statistically significant difference for it at either the $\alpha = .05$ or the $\alpha = .01$ significance levels.

The results in Table 9.1 on the following page, with respect to the 18 actions that were taken at both the $\alpha = .05$ and the $\alpha = .01$ significance levels, were identical on a row-by-row basis, to the actions that were taken for the other 11 combinations of dataset sizes and queries that were mentioned in Section 9.1. The only difference between the values in Table 9.1 and those in these other 11 tables were the p-values. So, instead of listing the information from all 12 tables, we use Table 9.1 as the representative for all 12 tables.

In the *reject* actions, the largest p-value among the 12 tables was 4.366×10^{-8}; for the

Table 9.1: Test Results for Kolmogorov-Smirnov test (two-tailed) for a test collection of 10 million synthetic documents and 100 unique randomly-generated queries.

ranking method	random var 1	random var 2	p-value	action $\alpha = 0.05$	action $\alpha = 0.01$
BC	ASL'	$\mathrm{ASL}'_{\mathrm{r}}$	$1.554e - 15$	reject	reject
	ASL'	$\mathrm{ASL}'_{\mathrm{g}}$	$1.554e - 15$	reject	reject
	$\mathrm{ASL}'_{\mathrm{r}}$	$\mathrm{ASL}'_{\mathrm{g}}$	1	fail to reject	fail to reject
CLM	ASL'	$\mathrm{ASL}'_{\mathrm{r}}$	1	fail to reject	fail to reject
	ASL'	$\mathrm{ASL}'_{\mathrm{g}}$	$1.458e - 13$	reject	reject
	$\mathrm{ASL}'_{\mathrm{r}}$	$\mathrm{ASL}'_{\mathrm{g}}$	$1.458e - 13$	reject	reject
DT	ASL'	$\mathrm{ASL}'_{\mathrm{r}}$	$1.554e - 15$	reject	reject
	ASL'	$\mathrm{ASL}'_{\mathrm{g}}$	$1.554e - 15$	reject	reject
	$\mathrm{ASL}'_{\mathrm{r}}$	$\mathrm{ASL}'_{\mathrm{g}}$	1	fail to reject	fail to reject
IDF	ASL'	$\mathrm{ASL}'_{\mathrm{r}}$	1	fail to reject	fail to reject
	ASL'	$\mathrm{ASL}'_{\mathrm{g}}$	$1.458e - 13$	reject	reject
	$\mathrm{ASL}'_{\mathrm{r}}$	$\mathrm{ASL}'_{\mathrm{g}}$	$1.458e - 13$	reject	reject
RC	ASL'	$\mathrm{ASL}'_{\mathrm{r}}$	1	fail to reject	fail to reject
	ASL'	$\mathrm{ASL}'_{\mathrm{g}}$	$1.458e - 13$	reject	reject
	$\mathrm{ASL}'_{\mathrm{r}}$	$\mathrm{ASL}'_{\mathrm{g}}$	$1.458e - 13$	reject	reject
WC	ASL'	$\mathrm{ASL}'_{\mathrm{r}}$	$1.554e - 15$	reject	reject
	ASL'	$\mathrm{ASL}'_{\mathrm{g}}$	$1.554e - 15$	reject	reject
	$\mathrm{ASL}'_{\mathrm{r}}$	$\mathrm{ASL}'_{\mathrm{g}}$	1	fail to reject	fail to reject

fail to reject actions, the p-value was 1. Inspection of the dataset contents revealed that, for all ranking methods, except for the DT ranking method, the reason that the p-value was 1 was because 100% of the value pairs had exact matches between the values that were being compared. In the DT case, there were 100% matches for 9 of the combinations of datasets and the values of the random variables being compared and 42.4-49% exact matches, and very small differences in the values being compared, for the remaining three combinations.

The first common theme to emerge from the information in the 12 tables was that there was always a statistically significant difference between the ASL' and $\mathrm{ASL}'_{\mathrm{g}}$ measures. The other common theme to emerge was that, for the other two comparisons (i.e., ASL' versus $\mathrm{ASL}'_{\mathrm{r}}$ and $\mathrm{ASL}'_{\mathrm{r}}$ versus $\mathrm{ASL}'_{\mathrm{g}}$) that are associated with a ranking method, there was a significant difference between the values that were being compared for equality by one of the comparisons but there were no significant difference between the values that were being compared for equality by the other comparison.

The intuitive reason for these themes is that there was a weak ordering between the accuracy of these ASL performance measure variants. The weak order can be expressed as

$$\mathrm{ASL}' \succeq \mathrm{ASL}'_{\mathrm{r}} \succeq \mathrm{ASL}'_{\mathrm{g}},$$

where $m_a \succeq m_b$ denotes that the absolute difference between the value for measure m_a and the value for $\mathrm{ASL}'_{\mathrm{g}}$ is at least as great as the absolute value of the difference between the value for measure m_b and the value for $\mathrm{ASL}'_{\mathrm{g}}$. This is related to the discussion on page 371 in Section 8.5 and, in particular, to Inequality 8.5.3 on page 372.

Due to this weak ordering, the greater absolute difference in the values is expected to occur between the values for ASL' and $\mathrm{ASL}'_{\mathrm{g}}$ rather than between the values for $\mathrm{ASL}'_{\mathrm{r}}$ and $\mathrm{ASL}'_{\mathrm{g}}$. Conversely, the smaller absolute differences are expected to occur between the values for ASL' and $\mathrm{ASL}'_{\mathrm{r}}$ or between $\mathrm{ASL}'_{\mathrm{r}}$ and $\mathrm{ASL}'_{\mathrm{g}}$.

The information in the immediately previous paragraph contains the reasoning behind there always being a statistically significant difference between the random variables ASL$'$ and ASL$'_{\mathrm{g}}$. This information is also the basis behind there being no statistically significant difference between the distributions for either the ASL$'$ and ASL$'_{\mathrm{r}}$ random variables or between the ASL$'_{\mathrm{r}}$ and ASL$'_{\mathrm{g}}$ ones, but not, both, for each ranking method.

9.3 Summary

This chapter showed the results of using the two-tailed version of the Kolmogorov-Smirnov test to analyze how well the ASL$'$, ASL$'_{\mathrm{r}}$, and ASL$'_{\mathrm{g}}$ measures compared with each other for performance prediction. Each of the 3 possible measure-to-measure comparisons had 3 random sets of queries, with different cardinalities, drawn from synthetic datasets of 4 different sizes. In total, the Kolmogorov-Smirnov test was run on 216 combinations of ranking methods (there were 6 ranking methods), measure-to-measure comparisons (there were 3 measure-to-measure comparisons), query sets (there are 3 query sets), and synthetic document sets (there were 4 synthetic document sets). If we multiply the number of categories for each of these 4 entities, we obtain $6 \cdot 3 \cdot 3 \cdot 4 = 216$.

The cardinalities of the sets of randomly generated queries was the same for each dataset. These cardinalities started at 100 and ended at 10,000. The synthetic dataset sizes started at 10 million documents and ended at 10 billion documents. There was an order of magnitude difference between the successive cardinalities for each set of queries. Similarly, there was the same order of magnitude difference between the cardinalities of successive sets of documents.

The analysis of the Kolmogorov-Smirnov test results corroborated the theoretical indications that there was a weak ordering between the three ASL variants that were being compared. The results analyses showed that there was statistically significant evidence that the ASL$'_{\mathrm{g}}$ measure was superior to the ASL$'$ one. The analyses also showed that, in some circumstances, there was no statistically significant difference between the ASL' and $\mathrm{ASL}'_{\mathrm{r}}$ measures. In other circumstances, the analyses showed that there was no statistically significant difference between the $\mathrm{ASL}'_{\mathrm{r}}$ and $\mathrm{ASL}'_{\mathrm{g}}$ measures. This was true for each ranking method, across all combinations of documents and queries. There was no ranking method where it was simultaneously true that there was a statistically significant difference between the comparisons of the ASL' and $\mathrm{ASL}'_{\mathrm{r}}$ measures and the comparisons of the $\mathrm{ASL}'_{\mathrm{r}}$ and $\mathrm{ASL}'_{\mathrm{g}}$ measures. From this, we can conclude that, there was no circumstance where the Kolmogorov-Smirnov test results, when taken as a whole, indicate that there was no statistically significant difference between the three measures when they were viewed as a whole.

Chapter 10

The ASL Measure and Three Frequently-Used Performance Measures

This chapter addresses the last of the three research questions that were enumerated in Section 3.5, which starts on page 103: When does the Average Search Length (ASL) performance measure and one of these measures (i.e., MZ-based E measure (MZE), Expected Search Length (ESL), Mean Reciprocal Rank (MRR)) both imply that one document ranking is better than another document ranking? More specifically, Section 3.5.3 contains the initial introduction for this research question. In order to help answer this question, equations were derived for these four performance measures, and the recall and precision measures, that were consistent with the assumption that the documents in a ranking may have duplicate (i.e., tied) retrieval status values (RSVs); then, data collections were developed to test these derivations; and, lastly, the results that were obtained by the use of these derivations were shown to be consistent with empirical results.

In addition to discussing the impact that duplicate RSVs have on document rankings and performance evaluation measures, this chapter also discusses these topics: graphical and analytic ways to compare two performance measures on the basis of how much they agree or disagree about the relative ranking of two document sequences; definitions of agreement and disagreement; an analytic way to help determine the amount of agreement and disagreement; characteristics to consider when comparing two measures; strong and weak orders; a general framework for determining the performance values for rankings that contain duplicate RSVs; combinatoric derivations for the Expected Search Length (ESL), Average Search Length (ASL), precision, recall, MZ-based E measure (MZE), and reciprocal rank (RR) performance measures; and the validation of these combinatoric-based derivations. This chapter concludes with two examples. The first example compares values that were generated by an ASL measure consistent with the assumption that documents may have tied (i.e., duplicate) RSVs with the ASL values for the best-case, random-case, and worst-case rankings for the same collection of documents. The last example compares the ASL measure with the MZE, ESL, and RR measures across 6 types of ranking methods.

The ultimate goal of the work that occurs in this chapter is to support the answering of the above research question. In order to be able to do this, it was necessary to create tools that allowed the ASL performance measure to be compared to the ESL, MZ-based E measure, and Mean Reciprocal Rank (MRR) performance measures. A major problem in comparing the results of several single value performance measures was that the resultant values were often incomparable across the measures. For example, does a value of 7.3 from the ASL measure for a query q, and a document collection of size 10,000, indicate worst performance, the same performance, or better performance than a value of 0.5 from the MZE measure for that same query and document collection? Is there a way to compare two measures when there may be no standard way to determine the relative goodness of one value on its scale of measurement with that of another one on a possibly different scale?

Another factor that must be considered when comparing performance measures is that a sequence of ranked documents may not be strongly-ordered due to the presence of duplicate RSVs. This was certainly the situation that was encountered in the different validations and analyses that were discussed in Chapters 8 and 9. Each of the 6 ranking algorithms produced only two distinct RSVs. As a consequence, the documents were clustered into just two groups. This clustering effect was attributed to the query-document model that was introduced on page 21 in Section 2.2.5 of Chapter 2. Namely, the model was concerned with two basic pieces of information: whether a document was relevant to the single term query and whether the document contained the query term. A direct result of these assumptions was that the model was very coarse-grained with respect to the ranking of documents. The ranked vectors of documents in this query-document model tended to have high numbers of duplicate RSVs because the RSVs in a ranking typically fell into one of these two categories: (a) zero and a positive value or (2) zero and a negative value. This resulted in many ties. The subject of ties and how to handle them are discussed in Section 10.5. Subsequent sections develop versions of the ASL, ESL, MZE, and several other measures that are consistent with the assumption that documents may have tied RSVs. Later, the values from these measures, that were consistent with the assumption that documents may have tied RSVs, were used to compare the performance of several of the measures at arbitrary points in a ranked vector V of documents.

10.1 Regions of Agreement and Disagreement About Relative Rankings

The method developed in Losee (2000) helped to answer the research question that was introduced in the first paragraph of this chapter. This method sidesteps the issue of the relative worth of values from two different measures. The Losee method works by directly comparing the value of each measure, at a particular point i from the front of a vector V of ranked documents, to the value of the same measure at another point j

in the same ranking. Then, for each measure, the signed difference (as contrasted with the absolute value of the difference) of the values of its two points are calculated. A difference of 0 means that the performance is the same at points i and j, a positive or negative value indicates that the performance changed. In the case of a change, the sign of the difference indicates the direction of the change. Assume that point i is no farther from the front of vector V than is point j. If these points are values on an ordinal scale, then this means that $i \leq j$.

Also, assume that the difference for a particular measure is always calculated by subtracting the value of the measure at point i from the value of the measure at point j. In order to determine whether the sign of a difference indicates better or worse performance, it is necessary to know whether higher values indicate better or worse performance. Lower values mean better performance for some measures (e.g., ASL, ESL, MZE) whereas higher values mean better performance for others (e.g., RR). The ASL-ESL-MZE category of measures was labeled as lower-is-better (LIB), the RR category of measures was labeled as higher-is-better (HIB).

The Losee (2000) method compares the relative performance of two measures by determining where they agree and disagree on the ranking of the documents in vector V at any two arbitrary points in the ranking. The signed differences are used to make this assessment. The non-negative function NN is used to transform each signed difference into either the Boolean value *true* or *false*. The performance interpretation of a *true* value, for an HIB type of measure, was that the performance at point j was as good, or better, than it was at point i. A *false* value for an HIB type of measure meant that the performance was worse at point j than it was at point i. The non-negative function NN is defined on a real number x as

$$\mathrm{NN}(x) = \begin{cases} \textit{true}, & \text{if } x \geq 0; \\ \textit{false}, & \text{otherwise.} \end{cases}$$

If two measures are both LIB types, or are both HIB types, then they are said to agree on the relative ranking between points i and j when their non-negative function signed difference values are the same Boolean value (i.e., both transformed differences are *true* or both are *false*); otherwise, they are said to disagree. If one measure is an LIB type and the other measure is an HIB type, then the signed difference of the LIB type of value must be negated before the non-negative function is applied to it. The reason is that erroneous results could occur because lower means better performance with the LIB type of measure whereas it means worse performance with the HIB type of measure. The negation has the effect of making both of the values HIB types. In other words, it effectively normalizes the directionality of "better-ness."

The pairs of points that correspond to disagreements between the two measures can be plotted on a graph to visually depict the regions of agreement and disagreement between the two measures. The points can be colored in an arbitrary, but consistent, way so that the white areas of the plots correspond to areas where the two measures agree on the relative rankings of the documents and the darker areas on the graphs correspond to the regions of disagreement. If the ranges of values for the measures being plotted are different, then it is advisable to normalize the ranges and plot both measures on axes where the values are in the closed interval $[0, 1]$.

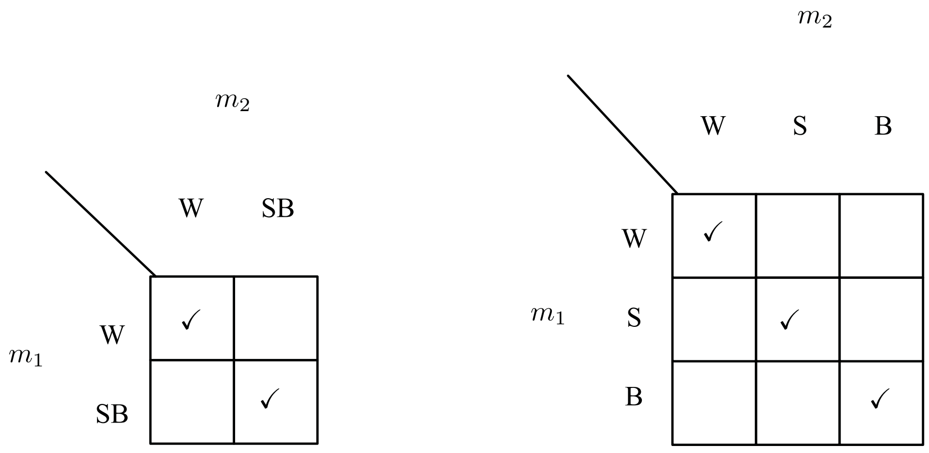
10.1.1 More Information About Performance Measure Disagreements

The method from Losee (2000) was extended in a straightforward manner to provide more information about the nature of any disagreements. The signed difference that is obtained from the values of a performance measure at points i and j provides information on whether the performance at point j is worse than (W), the same as (S), or better than (B) the performance at point i. The Losee method collapses the latter two categories into a single category: the performance at point j is the same as, or better than, the performance at point i. Let SB denote this new category. The extension of the method did not collapse any categories. It used the categories W, S, and B. Two measures were defined to be in agreement when their respective categories were equal; otherwise, they disagreed. Figure 10.1 shows all the possible ways that two measures m_1 and m_2 can agree and disagree with respect to their categories. The check marks on the main diagonals of the two subfigures illustrate when there is agreement between m_1 and m_2. The white squares illustrate where these measures disagree.

A key to acquiring this additional information was the use of the sign function rather than the non-negative function. The sign function is defined on a real number x as

$$\operatorname{sign}(x) = \begin{cases} -1, & \text{if } x < 0; \\ 0, & \text{if } x = 0; \\ +1, & \text{otherwise.} \end{cases}$$

The non-negative and sign functions can be also be viewed as mapping a real value x to a category. The non-negative function maps $x \geq 0$ to SB and $x < 0$ to W. Similarly, the sign function maps $x < 0$ to W, $x = 0$ to S, and $x > 0$ to B. Figure 10.1(a) shows that there are four possible combinations of mappings for two performance measures m_1



	W	SB
W	✓	
SB		✓

(a)

	W	S	B
W	✓		
S		✓	
B			✓

(b)

Figure 10.1: This figure details the categories of agreement and disagreement on relative levels of performance for measures m_1 and m_2 between two points i and j in a ranked vector V of documents, with point i occurring, before, or at, the same ordinal position as point j. For measure m_k, where $k \in \{1, 2\}$, the symbol W denotes that the performance at point j was worse than it was at point i. The symbol SB denotes that the performance at point j was either the same (S) as the performance at point i or that it was better (B) than the performance at point i. The check marks entries on the main diagonal in each subfigure indicate where the m_1 and m_2 performance measures agree on the relative rankings of two distributions of documents. The unmarked squares indicate where the two measures disagree on the relative rankings. The left half of this figure corresponds to the Losee method, the right half corresponds to the extended version of this method.

and m_2 with the non-negative function. Figure 10.1(b) on the previous page shows that there are 9 possible combinations of mappings for m_1 and m_2 with the sign function. The main diagonals in each of these subfigures show those combinations where m_1 and m_2 are in agreement; the squares that are not part of the main diagonals correspond to disagreements between the measures.

In Figure 10.1(a) on the preceding page, there are 2 combinations of disagreement. The top rightmost square corresponds to the situation where m_1 exhibits worse performance at point j than it does at point i whereas m_2 exhibits the same, or better, performance between these two points. The bottom leftmost square corresponds to the situation where m_1 exhibits the same, or better, performance at point j than it does at point i whereas m_2 exhibits that the performance at point j is worse than it was at point i. When there is a disagreement, if all that we are concerned about is whether one measure shows worse performance between these two points and the other one measure shows the same, or better, performance, then Figure 10.1(a) on the previous page, effectively, only has one type of disagreement.

Figure 10.1(b) on the preceding page shows 6 combinations of disagreements. Depending on one's perspective, there may be 3 or 6 types of disagreement. If it is not important to identify the measure whose performance decreased, stayed the same, or increased between point i and j, then, because the various combinations of disagreements are symmetrical around the main diagonal, there are, effectively, only three types of disagreement: (A) one measure exhibits worse performance at point j than it does at point i whereas the other measure exhibits the same behavior at point j than it does at point i; (B) one measure exhibits worse performance at point j than it does at point i whereas the other measure exhibits better behavior at point j than it does at point i; and (C) one measure exhibits the same performance at point j that it does at point i whereas the other measure exhibits the better behavior at point j than it does at point

i. However, if it is necessary to also indicate the measure and the change status, then 6 types of disagreement need to be indicated because measure identity information needs to be associated with each performance status.

The disagreement information for the plots in later sections of this chapter use 6 types of disagreement. The graphs for the Losee method typically consist of white regions (denoting agreement) and dark regions (denoting disagreement). The graphs that correspond to the extended method may consist of white regions (denoting agreement) and 6 other distinctly-colored regions to correspond to the 6 kinds of disagreement.

10.2 Characteristics to Consider When Comparing Measures

The comparison techniques in Losee (2000) require that, for any two measures being compared, there must be a way to calculate the values of these measures at any point in a ranking. This means that, for an N-document ranking, each measure's respective value must be calculable at any point k in the ranking where $1 \leq k \leq N$.

Before continuing further, it might be helpful to further clarify the notion of point that is used in much of the remainder of this chapter. Points are associated with a vector V of documents that are typically assumed to be ranked in non-ascending order by the retrieval status value (RSV) of their documents. In this context, a *point* is merely an index into V. That is, a point i in V denotes the index of the ith document from the front of V. For example, assume that vector V has 5 documents. Since V has 5 documents, it also has five points, or index positions. These points are numbered 1 through 5, inclusive, with point 1 corresponding to the index of the document at the front of V, point 2 corresponding to the index of the document next from the front of V, and so on, with point 5 corresponding to the index of the document at the end of vector V. The document at point i in V can be denoted as $V[i]$ and the RSV at this same point can be denoted as RSV$[i]$.

The notion of point that was just described in the immediately prior paragraph contrasts with another common notion of point, that is, where $\Pr(X = a)$ is the *point probability* (Terrell, 1999; Walpole, 2002) that the value of the random variable X is exactly a. Unless we state otherwise, the notion of point that we use in most of this chapter is that of an index position.

Two *point* measures (i.e., measures that can be calculated up to an arbitrary point in a ranking) may have different notions of what a point is. For example, the MRR and ESL are point measures because they are based on performance at a particular point in a ranking. A point for the MRR is an arbitrary position k from the beginning of a ranking whereas, for the ESL, a point is a user-specified number of relevant documents from the beginning of a ranking. Other measures, such as the MZE and ASL, are not point measures because they are based on the the *totality* of a ranking (i.e., *all of the points* in a ranking) rather than the performance at a *particular point* in a ranking.

A ranking V can be viewed as a *full* distribution D where the range of values for the points are integers in the closed interval $[1, \mathrm{card}(V)]$. Each of the $\mathrm{card}(V)$ (i.e., the cardinality of vector V) distinct points has a single RSV associated with it. The RSVs over the range of points that is covered by the interval are not necessarily all distinct, that is, the RSVs may all be the same value, they may all be different values, or they may be a mixture of different values. Rankings that are based on points x that do not cover the entire range of points, that is, $1 \leq x \leq b < \mathrm{card}(V)$, are analogous to what is known in the statistics literature as distributions that are *truncated from above*, or *right-truncated distributions* (Johnson et al., 2005). Distributions that use the full range of values in the closed interval $[l, u]$, where l is the lower bound of the range and u is the upper bound of the range, can be thought of as *full distributions.* Rankings that are based on the totality of the points for a vector V are full distributions, rankings that are based on points that begin at the first position in V and terminate at a position $1 \leq p < \text{card}(V)$ are right-truncated distributions. Virtually all of the distributions that we work with in the remainder of this chapter are right-truncated distributions.

Some other characteristics that must be taken into consideration when comparing measures are: Does the measure calculate values for a single query or a set of queries? Does the measure assume that the ranked documents are strongly ordered (i.e., each document has a unique retrieval status value)? Is the measure defined where there are no relevant documents in the ranking for a query q? Is the measure defined at every point in a ranking? When the value of the measure increases, does this indicate better or worse performance? Are the range of values the same for each measure? If not, do the values need to be normalized? The above characteristics play a major role later on when we extend and adapt the measures to fit into the framework provided by Losee (2000). Table 10.1 lists these characteristics for the ASL, ESL, MZE, and MRR measures. The next section contains a detailed discussion of these characteristics and their individual importances.

Table 10.1: Important Characteristics of the ASL, ESL, MZE, and MRR Performance Measures.

	measure			
characteristic	ASL	ESL	MZE	MRR
totality of ranking (T) or point (P)?	T	P	T	P
if point: fixed (F) or variable (V) position from front?	—	V	—	F
assumes that the ranking is strongly ordered?	N	N	Y	Y
single query (q) or set of queries (Q)?	q	q	q	Q
measure defined even when there are no rel. docs?	N	Y	N	Y
value of performance measure: lower is better (LIB) or higher is better (HIB)?	LIB	LIB	LIB	HIB
range of values	$[1, N+1]$	$[0, N]$	$[0, 1]$	$[0, 1]$

The previous paragraphs in this section discussed some important characteristics of the ASL, ESL, MZE, and MRR measures. Table 10.1 on the preceding page lists the values of these characteristics and facilitates comparing and contrasting them. The table shows that no two of the measures of interest have the same value for all of the corresponding characteristics. For example, some of the measures are point measures whereas other measures are totality ones. However, even the two measures that are point measures do not have the same notion of what a point is. It is important for comparison purposes that the the measures have the same values for their corresponding characteristics. One way to accomplish this is to decide on the desired value for each characteristic and then create similar measures from the original ones. Basically, the similar measures can be viewed as adaptations of the original ones.

10.2.1 Is the Measure Based on the Totality of a Ranking or on a Point In the Ranking?

This is the most important characteristic of the seven because the comparison techniques that are being used in this chapter require that the measures they use as arguments be defined for all of the ranks that are associated with a query q, and a ranking V, for a document collection of size N. These ranks correspond to physical positions starting at the front of a ranking. The first position in a ranking is numbered 1, the second position is numbered 2, and so on, with the last position being numbered N.

The above requirement can be handled by computing the value of a measure at an arbitrary physical position k in a ranking. It means that no matter how many documents there are in a ranking, the value of the performance measure is based only on the first k documents in the ordering. In the information retrieval (IR) literature, this truncation position is commonly referred to as *document cut-off at position k.*

MRR is naturally defined in terms of a cut-off position k and notationally is often written as MRR@$k(V)$. The MZE measure can very easily be defined in terms of a cut-off position because it is based on recall and precision. There are numerous references in the literature, especially with respect to Web searching, where both precision and recall are only defined for the first k documents in a ranking. Notationally, these variants are often expressed as $P@k(V)$ and $R@k(V)$, respectively. Hence, they can be used to define MZE@$k(V)$.

The Average Search Length is based on the totality of a ranking but can very easily be defined in terms of a cut-off value k. In the prior chapters of this dissertation, care was taken to always assume that there was at least one relevant document in the collection for a query q because, otherwise, the ASL would be undefined. When defining a version of the ASL, that can be calculated at an arbitrary document cut-off point k, one must be aware of the situation where, even though the collection has at least one relevant document for a query q, every ranking may not have a relevant document among some of its first k documents. Therefore, in addition to creating a version of the ASL, namely, ASL@$k(V)$, that can be calculated at various document cut-off points, a reasonable definition for this adapted measure must be provided when there are no relevant documents among the first k documents in a ranking.

The Expected Search Length can be viewed as being based both on the totality of a ranking, and also as being a point measure. The justification for it being a point measure is because the number of requested relevant documents x can be viewed as a point in a ranked vector V. Of course, it is a *relative* kind of point (rather than a *fixed* position k in a ranking) and can vary based on a query, document collection, and set of relevance judgments combination. The totality justification is due to the fact that the original version of this measure (i.e., ESL(V, x) does not specify a document cut-off point that is independent of the relevance of the documents in vector V. Notationally, the version of this measure, that can be calculated at an arbitrary document cut-off point, is denoted by ESL@$k(V, x)$, where k is the document cut-off point (as above) and x is the number of relevant documents to retrieve. In essence, ESL@$k(V, x)$ can be viewed as having two cut-off values: the document cut-off value k and the number of relevant documents x. The details of how these two cut-offs coexist and influence the measure calculations are discussed in Section 10.6.

With respect to the discussion in the previous paragraph, where one could argue that the ESL(V, x) measure has characteristics of both a point measure and a totality measure, it was treated as a point measure in this dissertation and defined as being equivalent to the document cut-off version with the cut-off values being the same as the cardinality of vector V, that is,

$$\mathrm{ESL}(V, x) \equiv \mathrm{ESL@}c(V, x),$$

where $c = \mathrm{card}(V)$.

10.2.2 Does the Measure Assume That the Ranked Documents Are Strongly Ordered?

Many IR performance measures assume that ranked documents are strongly ordered. Section 10.3 contains a detailed discussion of strong and weak orders. When a ranking is not strongly ordered, these measures may compute incorrect values because they are not sensitive to the presence of *ties.* Ties arise when there are two or more documents in a ranking that have the same RSV. In such a ranking, it is possible that there may be more than one distinct RSV with each of them having a set of several documents that are associated with it. When the value for a measure is not computed correctly, the true value is either underestimated or overestimated. The differences may be such, that when comparing how well several measures perform, the relative ranking of these measures can be affected if the underestimation or overestimation is significant enough.

Implications for Performance Evaluation

The assessment of how well a ranking algorithm works in a particular situation is typically handled by performance evaluation software such as trec_eval (Buckley and Voorhees, 2005; Voorhees and Harman, 2005; Voorhees, 2005) and inex_eval (Vu and Gallinari, 2005) that applies one or more performance measures to the output of the ranking software. At a minimum, the inputs to performance evaluation software usually consist of a query q and the associated vector of ranked documents. Often, the input also includes the RSV for each rank and a unique document identifier for each document..

> Typically these performance measures assume that a ranking algorithm arranges the results of a query into a total ordering, *i.e.* no two results to a query have the same [RSV]. This assumption is reasonable for scoring functions that map a rich set of features of the result document to a real-valued score, but it is less warranted for evaluating the performance of a single discrete feature, *e.g.* page in-degree, click count, and page visits. (McSherry and Najork, 2008)

The comments in the quoted passage above are particularly germane, not just to the performance evaluation efforts in this dissertation, but to IR performance evaluation in general. Their particular relevance to this dissertation is due to the statement in Section 4.1 that "[t]wo essential characteristics of the [performance] models [that are used in this dissertation] are binary relevance and that the single query term is either present or absent in a document." A consequence of these characteristics has been that the ranked lists that were generated by use of the feature weights in Table 7.1 on page 329, and the query and document weight relationships that were enumerated in Figure 7.9 on page 329, is that the result is ranked lists that contain a maximum of two distinct RSVs. Hence, a ranking contains large numbers of documents that have the same RSV.

The IR literature indicated that a common approach to handle this tie problem was to break ties arbitrarily. This was often done in one of two ways: randomly select one of the valid sequences or use a document identifier (like what is done in TREC) as the tie-breaker. For this dissertation, there were problems with both of these ways. First, the ordering was nondeterministic with random selection. Second, the document identifier as a tie-breaker had at least two drawbacks: (1) multiple documents in a collection could have the same identifier and (2) several of the documents in a collection may not have a document identifier. Neither the random selection approach, nor the document identifier approach, provided a guarantee against either underestimation or overestimation of the value of the performance measure.

A better approach was to base the value of the performance measure on the average performance over all of the possible document orderings or sequences with each sequence considered to be equally likely. This is the approach that was taken in this chapter. This manner of calculation was not only defendable from a statistical and probabilistic viewpoint, but it also had another desirable quality – the value of the performance measure was always deterministic.

An Example of the Estimation Problem

This example illustrates the essence of the estimation problem. Suppose we have a query q, a document collection of size 3, and the associated relevance judgments. One document is labeled A, another is labeled B, and the remaining one is labeled C. Documents A and B are relevant to query q, but document C is not relevant to the query. Assume that the query q, the three documents, and the set of relevance judgments are input to ranking software that produces as its output the ranked list of these documents, along with the RSV at each rank.

If the RSVs of all documents are pairwise distinct (i.e., no two documents have the same RSV), then the ranked list always corresponds to one of the 6 sequences in Table 10.2 on the next page. Without loss of generality, let the output of the software that implements the ranking algorithm be that of Sequence 3 (i.e., Document B is ranked first, Document A is ranked second, and Document C is ranked third). The evaluation algorithm calculates that the MSL for this sequence is 3/2. Now, assuming that the query, the document collection, the set of relevance judgments, and the ranking software remain the same, the software always generates the ranking that corresponds to Sequence 3 no matter how many times it generates rankings for the fixed set of inputs. The output of the ranking software is deterministic for the scenario that was described in this paragraph.

In the previous paragraph, we considered the impact that pairwise distinct RSVs had on the stability of a ranking for certain fixed factors. In this paragraph, we consider the opposite end of the spectrum — the RSV for Document A is v and the RSVs for the other two documents are also v (i.e., the RSV is the same for every document). In this case, all that can be guaranteed from one run of the ranking software to another run for this fixed set of inputs is that the document ranking corresponds to one of the 6 sequences that are listed in Table 10.2. Multiple runs of the ranking software may produce a different document sequence each time the software is run. In other words, the output of the ranking software may be nondeterministic because each of the 6 sequences is a possible output candidate.

Table 10.2: The MSL and ASL of All Possible Sequences of Two Relevant Documents (A & B) and One Non-relevant Document (C) When All Three Documents Have the Same RSV.

	rank				
sequence	1	2	3	MSL	ASL
1	A	B	C	3/2	2
2	A	C	B	2	2
3	B	A	C	3/2	2
4	B	C	A	2	2
5	C	A	B	5/2	2
6	C	B	A	5/2	2

Assume that we wish to compute the ASL from the ranked documents. The effect of all three documents having the same RSV, with respect to this example, on the calculation of the ASL is evidenced in Figure 10.2 on page 411 by the variability of the value for the Mean Search Length (MSL) measure. If adjustments are not made to eliminate the nondeterminism, then the reported ASL is always nondeterministically equal to the MSL value of one of the 6 sequences.

As it was introduced and defined in a previous chapter, the MSL is specific to an individual ordering and is calculated by totaling the positions of the relevant documents in an ordering and then dividing that quantity by how many relevant documents there are in the ordering. Table 10.2 on the preceding page contains the MSL value for each of the possible sequences in our example. For the convenience of the reader, we restate that, in Section 7.10.2, the MSL and ASL were shown to be closely related. In fact, for this example, the ASL can be obtained by calculating the average of the MSL values. That is,

$$\text{ASL} = (3/2 + 2 + 3/2 + 2 + 5/2 + 5/2) = 12/6 = 2.$$

Figure 10.2 on the next page illustrates the variability among the 6 MSL values that were generated from the data in Table 10.2 on the preceding page. If the evaluation algorithm assumes that the sequence of documents is strongly-ordered by the RSVs, then one of the 6 non-distinct MSL values is the value that it calculates for the ASL. But, if all the documents actually have the same value for their respective RSVs, then the value that an evaluation algorithm calculates for the ASL is the mean of the 6 MSL values – provided that the algorithm used is consistent with the assumption that documents may have tied RSVs. This same ASL value is the one that is calculated no matter which of the 6 possible sequences the ranking algorithm places the documents in according to their RSVs. In our example, Figure 10.2 on the next page shows that the ASL value is 2 for all of the 6 sequences even though the MSL value for a sequence may not necessarily be 2.

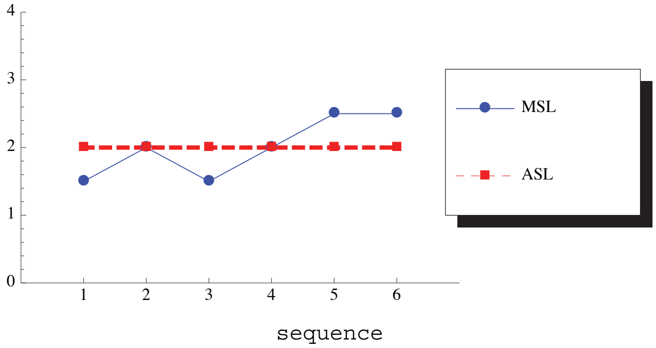


Figure 10.2: A Line Plot of the MSL and ASL From the Data in Table 10.2 on page 409.

10.2.3 Is the Measure Based on a Single Query?

This chapter compares how well single queries perform rather than a set of queries. According to the information in Table 10.1 on page 403, all of the four measures listed there, with the exception of the MRR measure, are single query measures. The MRR measure can be transformed into a single query measure by restricting the cardinality of its set of queries to 1 (i.e., the set is a singleton set). With this restriction, the Mean Reciprocal Rank measure effectively becomes the Reciprocal Rank measure .

10.2.4 Is the Measure Defined Even When There Are No Relevant Documents?

The ESL and RR measures are defined even when there are no relevant documents in vector V. By contrast, the ASL and MZE measures are undefined when vector V has no relevant documents.

ASL

What is the appropriate value for the ASL when there are no relevant documents? Should it be assigned a value of 0? Or should it be assigned the value at the lower end (i.e.,

1) or the higher end of the range (i.e., N) for a document collection of size N? The 0 value is not appropriate because lower ASL values indicate better performance than higher ASL values. Neither of the other two assignments seem appropriate either because they correspond to a sequence that has one relevant document, and that document is either the first or last one in the sequence, respectively. The author feels that because there is some cost associated with the examination of a sequence that does not have any relevant documents, the assigned value should be at the higher end of the range, but should *not* be a valid value within the range. Due to these considerations, the author felt that a reasonable way to handle a no-relevant-documents sequence in this research was to assign a value of $N + 1$ to the ASL. This decision was incorporated into the performance evaluation model that was used in this dissertation for the ASL.

A way to conceptualize this decision is to imagine that each sequence of N documents has a virtual relevant document associated with it. That document always occupies a position that is one past the end of the sequence. That is, it is at position $N+1$. For ASL computation purposes, this virtual document only enters into the computation when all of the prior N documents are non-relevant. In other words, if a sequence has at least one relevant document, the virtual (and $N + 1$st) document does not play any role in the computation of the ASL value.

MZE

This measure can be easily modified so that is is well-defined even in the absence of any relevant document(s) in V (McSherry and Najork, 2008). The definition used in this dissertation is the one that was developed by McSherry and Najork (2008).

10.2.5 Does an Increase in the Measure's Value Correspond to an Increase in Performance?

The information in Table 10.3 indicates that lower values for the ASL, ESL, and MZE measures indicate better performance than do higher values. However, this is not the case with the RR measure. The opposite is true in that higher values for it indicate better performance than lower values. These different performance directions must be accounted for when the measures are compared later in this chapter.

Table 10.3: Important Characteristics of the Extended and Adapted Versions of the ASL, ESL, MZE, and MRR Performance Measures.

	measure			
	ASL	ESL	MZE	RR
characteristic	$@k(V)$	$@k(V, x)$	$@k(V)$	$@k(V)$
totality of ranking (T) or point (P)?	P	P	P	P
if point: fixed (F) or variable (V) position from front?	F	F & V	F	F
assumes that the ranking is strongly ordered?	N	N	N	N
single query (q) or set of queries (Q)?	q	q	q	q
measure is defined — even when there are no relevant docs?	Y	Y	Y	Y
value of performance measure: lower (LIB)/higher (HIB) is better	LIB	LIB	LIB	HIB
range of values	$[1, k+1]$	$[0, k]$	$[0, 1]$	$[0, 1]$

10.2.6 Do the Measures Use the Same Range of Values to Report Performance?

Generally, the answer to this question is going to be "no" when investigating the output from a collection of IR performance measures. The information in Table 10.3 indicates that it is also "no" for the four measures in it. The typical way to handle measures that have different ranges of values to assess performance is to normalize these ranges. This normalization approach is the one that is used later in this chapter.

10.3 Weakly and Strongly Ordered Rankings

Essentially, a *ranking* of entities (e.g., documents, humans, SAT scores) is an ordered sequence of entities. The particular sequence is largely a function of the ranking function and the values of one or more designated attributes (*ordering variable(s)*) associated with these entities. For a collection of documents, there is likely only a single ordering variable (the *Retrieval Status Value* (RSV)); for a human, there might be two ordering variables (e.g., grade, height) if, say, we wanted to rank students first by their SAT score and then by height within the score group.. Note that the concept of an RSV is discussed in much more depth in Section 7.9 (Retrieval Status Value, Weights, and Document Ranking).

Typically, the goal of a ranking endeavor is to place entities into either an ascending or descending sequence based on the values of their ordering variable(s). However, this is not always possible due to the possibility that two or more of the entities being ordered may have identical values for their ordering variables. In this case, the best that we can do is to place these entities into non-descending and non-ascending orders, respectively. If all of the entities have distinct (i.e., unique) values associated with their ordering variables, then the non-descending order would also be an ascending order. Similarly, a non-ascending order would also be a descending order.

Let n represent the number of ordering variables for a specific ranking and collection of entities. Then the ordering variables form an n-tuple where the parts, starting at 1, and ending at n, without skipping any parts, form a *sort key.* Without loss of generality, assume that part 1 is the major part of the key, that part 2 is the next most major part of the key, and so on, with part n being the least major part of the key. The least major part of a key is also often called the *minor* part of a key. If there is only one ordering variable, like with the RSV, then this is as simple as a sort key can be. The solitary ordering variable is both the major and minor part of the sort key. On the other hand, if there are multiple ordering variables, then these variables must be placed in a key in the order that is harmonious with the roles that the parts play, that is, the most important variable, for ranking purposes, should correspond to part 1, the next most important variable should correspond to part 2, and so on. For example, the sort key would be $(\text{SATscore}, \text{height})$ for the example that was just mentioned in the first paragraph of this section.

Assume that we have two arbitrary n-tuples

$$K_i = (p_{i,1}, p_{i,2}, p_{i,3}, \ldots, p_{i,n},)$$

and

$$K_j = (p_{j,1}, p_{j,2}, p_{j,3}, \ldots, p_{j,n},)$$

that represent the sort key values (Ks) for any two arbitrary entities of a collection C of size N where $n \in \mathbb{Z}^+$.

Informally, the entities in C are said to be *strongly ordered* by the $<$ (less than) relation if, for all $i \neq j$, the following assertion is true: $K_i < K_j$ implies that $p_{i,x} < p_{j,x}$ for at least one value of x in the range 1 to n, inclusive, and for all $y < x$, the assertion $p_{i,y} \leq p_{j,y}$ is true. Similarly, the entities can be said to be strongly ordered by the $>$ (greater than) relation if, for any $i \neq j$, the following assertion is true: $K_i > K_j$ implies that $p_{i,x} > p_{j,x}$ for at least one value of x in the range 1 to n, inclusive, and for all $y < x$, the assertion $p_{i,y} \geq p_{j,y}$ is true.

Strong ordering implies that each of the entities being ordered has a unique sort key. If this is not so, then at least two entities have the same sort key values and, hence, duplicate sort key values are present. In this case, we have a *weak order.*

Notationally, let $a \prec b$ represent that entity a comes before entity b in a ranking (i.e., the rank of entity a is a lower value than the rank of entity b); let $a \succ b$ represent that entity a comes after entity b in a ranking (i.e., the rank of entity a is a higher value than that of entity b); let $a \preceq b$ represent that entity a ranks the same as, or lower than, entity b; and let $a \succeq b$ represent that entity a ranks the same as, or higher than, entity b. The first two notations represent strong orders, the last two represent weak orders.

As an example, in Table 10.4, there is a strong order on the values for *rank*, e.g.,

$$1 \prec 2 \prec 3 \prec 4 \prec 5 \prec 6 \prec 7 \prec 8 \prec 9 \prec 10 \prec 11 \prec 12 \prec 13 \prec 14 \prec 15 \prec 16 \prec 17$$

and a weak order on the values for RSV, e,g.,

$$12 \succeq 12 \succ 9 \succ 3 \succeq 3 \succeq 3 \succ 2 \succeq 2 \succeq 2 \succ 1 \succeq 1 \succeq 1 \succeq 1 \succeq 1 \succ 0 \succeq 0 \succeq 0.$$

Note that the order is weak for the latter sequence because that sequence has at least one instance of $\succeq$ that had to be used to relate its ranked entities. Using terminology from statistics, the RSV variable in this table can be viewed a factor with six *levels* (i.e., unique values), namely, 12, 9, 3, 2, 1, 0. Each of these factor levels, except for the one that is associated with the value 9, has multiple entities associated with it. Figure 10.3 illustrates these levels and their associated entities.

Table 10.4: Ranked List of Seventeen Documents (R=relevant).

rank	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*
RSV	12	12	9	3	3	3	2	2	2	1	1	1	1	1	0	0	0
relevant?		R	R		R		R			R	R	R		R		R	

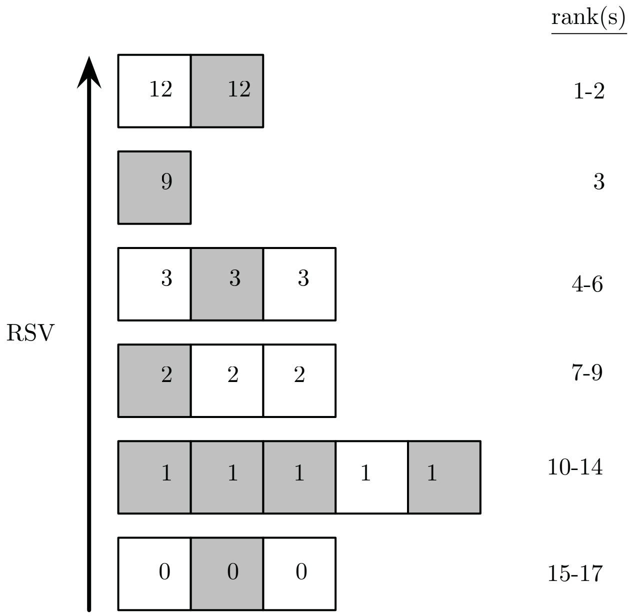


Figure 10.3: These are the 6 levels of the RSV factor from Table 10.4. The top level corresponds to the documents at ranks 1 and 2; the bottom one corresponds to the documents at ranks 15, 16, and 17. The gray boxes represent relevant documents whereas the white ones represent non-relevant documents. The RSV is at its maximum at the top level and is at its minimum at the bottom level. The ranking algorithm considers the documents at higher levels to be better satisfiers of the information need expressed by a query q than any of those at lower levels. Within a level, it considers all the documents at *that* level as being equal satisfiers of the information need. In other words, there is no significance to the position that a document occupies at a particular level.

10.3.1 What Does "Rank" Mean When Entities Are Weakly Ordered?

When N entities can be strongly ordered, the concept of rank is unambiguous because no entity has the same sort key value as any other entity in this order. In this case, the number of factor levels equals N, each factor level has exactly one entity associated with it, and each entity has exactly one factor level associated with it. In essence, there is a one-to-one correspondence (i.e., bijection) between the set of factor levels and the set of entities.

But, what is the rank when two or more sort key values are the same? In this situation, the only order possible is a weak order and is a more complicated situation than when the sort key values are distinct (which results in a strong order). Both Table 10.4 on page 416 and Figure 10.3 on the previous page illustrate the weak ordering of entities.

10.3.2 Nondeterministic Rankings

Without loss of generality, let the output of one run of a ranking algorithm be the sequence of documents that appear in Table 10.4 on page 416. In this table, the documents with the highest RSVs occupy ranks that are labeled 1 and 2. These documents consist of one that is relevant and one that is non-relevant. The non-relevant document is at rank 1 and the relevant document is at rank 2. As is typical with IR ranking algorithms, documents are ranked in reverse order of their RSVs. This is also true for the sequence in this table.

Suppose the ranking algorithm is run again with the same input(s), that is, the same query and the same number N of documents. This time, though, assume that it is the relevant document that is now at rank 1. This means that the non-relevant document is now at rank 2. Which sequence of documents is correct? Is it the sequence that is depicted in Table 10.4 on page 416, or is it the sequence that was obtained from running the ranking algorithm the second time? The answer is that both are correct because, in general, there is no guarantee that a ranking algorithm retains the same relative ordering among documents that have the same RSV value. The reason for this non-deterministic behavior has both practical and theoretical explanations. The practical explanation is discussed first.

In IR systems, the ranking algorithm is typically effected by sorting the documents according to their RSV values, but other techniques are also commonly used. As long as the ranking algorithm separates documents into groups based on their retrieval status values (i.e., RSVs), and sequences (i.e., orders) these groups such that the particular ordering relation holds, any of possibly multiple, but equivalent, sequences are possible.

As an example, assume that a sort-based ranking algorithm is used and that the sort key is the RSV. If the sort has been implemented correctly, the ranking algorithm uses the sort key value of each document to place these documents into a non-ascending order. Even if the query and the document collection are constant from one ranking request to the next one, the sequence of documents may be different for any of several reasons. One possibility for non-deterministic ranking behavior is that the underlying sort algorithm may not guarantee that the sort output is *stable* (i.e., documents that have the same RSV retain the reverse of their relative input order after the sort has taken place). Another possibility is that the unsorted documents may have a different input sequence from one run to the next. A third possibility has to due with memory, disk space, and buffer size; some sorting algorithms are more sensitive to these than others. Finally, a fourth possibility has to do with parallelism – the sort may be multi-threaded rather than single-threaded.

10.3.3 Smoothing for Nondeterministic Rankings

On the theoretical side, consider the fact that a particular level of the RSV factor has, say, m documents of which r are relevant. Since the documents at this level all have the same RSV, they can be permuted into

$$\frac{m!}{r!(m-r)!} = \binom{m}{r}$$

distinct sequences because there are r relevant documents and $m - r$ relevant ones. Since each of these sequences are equivalent, at least as far as the ranking algorithm is concerned, and absent any evidence that one of these sequences is more likely than any of the others, we must average the performance metrics for this level in order to obtain a truer value of this metric. These considerations become very important when we discuss the formulas for the Average Search Length, the Expected Search Length, the MZE measure, and the Reciprocal Rank measures.

10.4 Several Sum and Binomial Identities

The parameters $a, b, k, l, m, n, r, s,$ and q that appear in the identities below are all assumed to be integers. These identities represent the ones that are repeatedly applied in many of the derivations that take place later in this chapter.

Each identity is presented along with the parameter constraints that apply to their use in a formula or derivation. The main reference sources for these identities were Graham et al. (1994), Purdom and Brown (1985), Benjamin and Quinn (2003), and Larsen (2007). Most of the names that are used for these identities came from Graham et al. (1994).

10.4.1 Manipulation of Sums

These three identities are from Graham et al. (1994). The first one enables a quantity c, whose value is independent of the summation variable k, to be moved outside the summation. The second identity enables a single summation to be broken up into two independent summations. The third identity states that the sum of the a_k quantities is equal to the sum of any permutation $p(k)$ of the a_k quantities.

$$\sum_{k\in K} ca_k = c\sum_{k\in K} a_k. \qquad \text{(distributive law)} \tag{10.4.1}$$

$$\sum_{k\in K}(a_k + b_k) = \sum_{k\in K} a_k + \sum_{k\in K} b_k. \qquad \text{(associative law)} \tag{10.4.2}$$

$$\sum_{k\in K} a_k = \sum_{p(k)\in K} a_k. \qquad \text{(commutative law)} \tag{10.4.3}$$

10.4.2 Basic identities

For $n \geq 0$,

$$\binom{n}{k} = \begin{cases} 0, & \text{if } k < 0 \text{ or } k > n; \\ 1, & \text{if } k = 0 \text{ or } k = n; \\ n, & \text{if } k = 1; \\ n(n-1)\cdots(n-k+1)/k!, & \text{if } 2 \leq k \leq n. \end{cases} \tag{10.4.4}$$

Equation 10.4.4 could easily be viewed as four separate identities. They are being presented here as just one identity, though, because they are such simple and basic identities that the author of this dissertation strongly believed that they should be consolidated in one place.

The first line of this identity states that the number n distinct entities, chosen k at a time, without regard to order, is 0 if the number of entities that is chosen is negative (this is an impossible situation) or if the number chosen is greater than the number (i.e., n) that is available (another impossible situation). The second line of the identity states that there is only one way to either choose none, or all, of the entities. The third line states that the number of ways that one entity can be chosen at a time is the same as the number of distinct entities that are available. Lastly, the fourth line of this identity states that the number of ways to choose $k \geq 2$ entities at a time can be accomplished by calculating the value of n (the number of entities) times the product of the first $k-1$ integers that are smaller than n, and then dividing this value by $k!$ (the number of ways that k distinct entities can be permuted). For example, if $n = 10$ and $k = 4$, the product of n and the next $k-1$ smaller integers is

$$n(n-1)\cdots(n-(k-1)) = 10 \cdot 9 \cdot 8 \cdot 7 = 5040.$$

For $n \geq 0$,

$$\binom{n}{k} = \begin{cases} \frac{n!}{k!(n-k)!}, & \text{if } 0 \leq k \leq n; \\ 0, & \text{otherwise.} \end{cases} \tag{10.4.5}$$

This identity expresses the number of n distinct entities, chosen k at a time, in terms of factorials. The second line of this identity states that is impossible to choose more entities k than the number n that is available.

10.4.3 Symmetry

For $0 \leq k \leq n$,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \binom{n}{n-k}. \tag{10.4.6}$$

The entities to be chosen k at a time can be viewed as belonging to two distinct categories, with k of them belonging to one category and the remaining $n-k$ belonging to the other category. This relationship means that k and $n-k$ can be used interchangeably in the "choose" (i.e., bottom) part of the binomial. This identity is often used to transform the "choose" part to a simpler, or more easier to manipulate, form.

10.4.4 Addition

For $0 \leq k \leq n$, except when $n = 0$ and $k = 0$,

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}. \tag{10.4.7}$$

This identity enables the expression of one binomial as the sum of two other similar binomials provided that certain basic conditions are met.

10.4.5 Convolution identities

The following identities, under certain conditions, of course, allow us to simplify binomial expression manipulations by replacing a sum of binomial products with just a binomial.

For $m \geq 0, n \geq 0:$

$$\binom{m+n}{k} = \sum_{j=0}^{k} \binom{m}{j}\binom{n}{k-j}. \tag{10.4.8}$$

$$\binom{r+s}{m+n} = \sum_{k} \binom{r}{m+k}\binom{s}{n-k}. \tag{10.4.9}$$

For $l \geq 0, m \geq 0$, and $n \geq q \geq 0$,

$$\binom{l+q+1}{m+n+1} = \sum_{k=0}^{l} \binom{l-k}{m}\binom{q+k}{n}. \tag{10.4.10}$$

For $n \geq 0$, and $0 \leq m \leq l$,

$$\binom{l+q+1}{m+n+1} = \sum_{k=-(q-n)}^{l-m} \binom{l-k}{m}\binom{q+k}{n}. \tag{10.4.11}$$

10.4.6 Sum of the first n positive integers

For $n \geq 0$,

$$\frac{n(n+1)}{2} = \binom{n+1}{2} = \sum_{j=1}^{n} j. \tag{10.4.12}$$

This identity is one of the most well-known ones in discrete mathematics. It is an identity that many people who use discrete mathematics and combinatorics learn very early in their study of basic summations. We make use of it many times in the derivations that occur in the remainder of this chapter. Note that, by the definition of a binomial, the sum above can be expressed combinatorially as $\binom{n+1}{2}$.

10.4.7 Sum of several natural numbers

For $a \geq 0$, $b \geq 0$, and $a \leq b$,

$$\sum_{j=a}^{b} j = \sum_{j=1}^{b} j - \sum_{j=1}^{a-1} j = \frac{b(b+1)}{2} - \frac{(a-1)a}{2}. \tag{10.4.13}$$

This identity is useful when it is necessary to determine the sum of natural numbers in the closed interval $[a, b]$. The identity that is represented by Equation 10.4.12 is a special case of this identity.

10.4.8 Absorption identities

The “choose” (i.e., bottom) part of a binomial term, that appears in a summation, often has a dummy index variable k as part of the expression that resides there. This index term may appear at multiple places in the expression that is being summed. These multiple instances make summations more difficult, sometimes in a very complicated manner. If some of these instances can be removed by “absorbing” them into a nearby binomial term, this absorption can make manipulation of the entities in the summation vastly easier. Absorption can be observed in the three identities below. Notice that, in all three identities, the expression on the right-hand side of the equals sign has one less instance of k than the corresponding expression on the left-hand side.

For $k \neq 0$,

$$\frac{r}{k}\binom{r-1}{k-1} = \binom{r}{k}. \tag{10.4.14}$$

For $0 \leq k \leq r$:

$$k\binom{r}{k} = r\binom{r-1}{k-1}. \tag{10.4.15}$$

$$(r-k)\binom{r}{k} = r\binom{r-1}{k}. \tag{10.4.16}$$

10.5 A General Framework For Handling Ties

Cooper (1968) appears to be the earliest reference in the IR literature that acknowledged the presence of ties and suggested a way to adjust for them in the model for a performance measure. This acknowledgement and adjustment for ties appeared in his 1968 article on the Expected Search Length (ESL). The ESL measure does not assume that a ranked vector V of documents is strongly-ordered. This measure works correctly with both weakly- and strongly-ordered sequences of documents. The second earliest reference in the IR literature to a measure that incorporates the possible presence of ties in its calculations appears to be the precall measure (Raghavan et al., 1989).

In the first paragraph of this chapter, we remarked that some document rankings may contain tied, or duplicate, RSVs and that some performance measures calculate values that are consistent with the assumption that documents *may* may have tied, or duplicate, RSVs. In Section 10.2.2, we noted that some performance measures calculate values that are consistent with the assumption that the RSVs in a document ranking are distinct. These two types of assumption are mentioned many times in the subsequent pages of this chapter. In order to help with the economy of expression for measures that calculate values under these assumptions, we use *Type-T* as an adjective to denote a measure whose calculated values are consistent with the assumption that some of the documents in a vector V of ranked documents *may* have tied (i.e., duplicate) RSVs and use *Type-D* as an adjective to denote a measure whose calculated values are consistent with the assumption that all the documents in a vector V *must* have distinct RSVs. Most of the discussions in the remainder of this chapter involve Type-T performance measures rather than Type-D ones.

Some of the more recent literature that discusses the presence of ties, the need to handle them, and the development of either new measures to accommodate them, or how to adapt some of the most used measures (e.g., MRR, precision, recall, NCDG) to handle them, are Chiu et al. (2008), Lin et al. (2008), and McSherry and Najork (2008).

The tie-handling framework that is used in the remainder of this chapter comes from the McSherry and Najork (2008) article. Also, this chapter uses some of the notation and terminology from this article. Figure 10.4 on the following page introduces several of the most important concepts and notation.

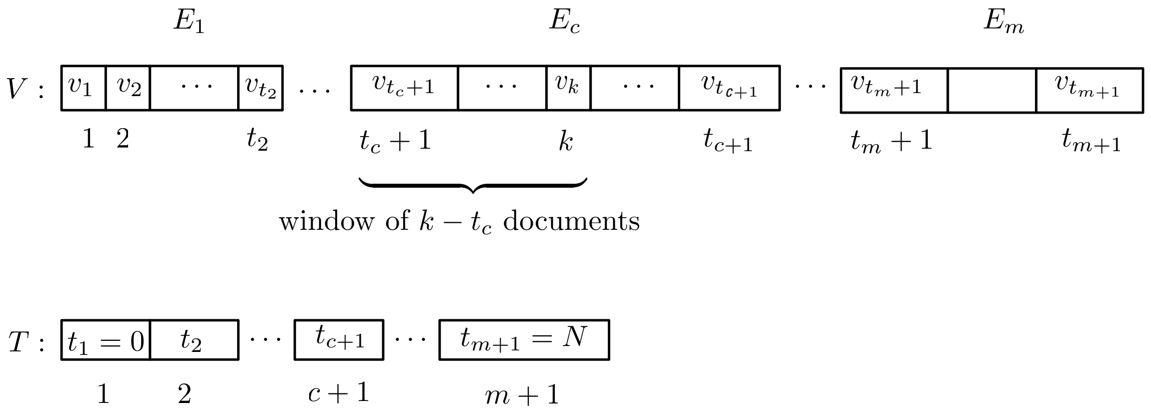


Figure 10.4: This diagram details the relationship between V (the vector of ranked documents) and T (the tie vector) for a document collection of size N. It depicts a situation where V has m equivalence classes labeled $E_1, E_2, \ldots, E_m$. The first $k - t_c$ positions in subvector V_c comprise the window for document cut-off k. The tie vector T has $m + 1$ members and its last m members contain the indices of the last element in each of the equivalence classes in V. Its first element is special and has the value of 0. The last element in T is equal to N.

The framework assumes that a *vector of ranked documents* V exists that has N documents non-ascendingly ordered by their RSVs. If no two documents have the same RSV, then the non-ascending order is effectively a descending order. All documents that have the same RSV belong to the same equivalence class. A ranking can have as few as one equivalence class (i.e., all the documents have identical RSVs) or as many as N equivalence classes (i.e., all the RSVs are distinct). The classes are labeled $E_1, E_2, \ldots, E_m$ where m is an integer that ranges from a low value of 1 to a value that can be at most N, the number of documents in the collection.

The indices for vector V range from 1 to N, inclusive. The first element in V is at index 1, the last element is at index N. The value of the first element in V is denoted by v_1, the value of the second element is denoted by v_2, and the value of the last element is denoted by v_N. It is assumed that vector V has $m \geq 1$ equivalence classes.

There is a *tie vector* T that is associated with vector V. This vector has $m+1$ elements and contains the indices of the last element of each equivalence class. The indices for vector T range from 1 to $m+1$, inclusive. The reason that vector T has $m+1$ elements instead of just m elements is due to its first element having the value 0 (this helps to simplify some of the computations that are used later to reason about ties). The value of the first element of T is denoted by t_1 (it always has the value 0), the value of the second element is denoted by t_2 (it is the index of the last element in the part of V that corresponds to the first equivalence class), and the index of the last element in T is denoted by t_{m+1} (it always has the value N).

Notationally, let r_i and n_i denote the number of relevant and total number of documents, respectively, in E_i. Let

$$V_i = <v_{t_i+1}, v_{t_i+2}, \ldots, v_{t_{i+1}}>$$

denote a *subvector* of the elements in V. Additionally, let R_i and N_i denote the number of relevant and total number of documents, respectively, that precede subvector V_i in V. Finally, let the indicator function I_R return a value of 1 if the document that is associated with v_i is a member of the set R of relevant documents for a query q and return a value of 0, otherwise. That is,

$$I_R(v_i) = \begin{cases} 1, & \text{if } v_i \text{ is a member of the set of relevant documents } R \text{ for a query } q; \\ 0, & \text{otherwise.} \end{cases}$$

This framework also includes the notion of a *document cut-off at index* k in the ranked vector V of documents. The notation V_c denotes the subvector of V that has k as the index of one of its elements. This subvector is also known as the document cut-off vector and corresponds to equivalence class E_c. The cut-off window that the kth document is a part of has $k - t_c$ elements and these elements occupy the first $k - t_c$ slots of V_c. Overall, this subvector has n_c elements of which r_c are relevant. The value of the kth element in V is denoted by v_k, where k is an integer in the half open interval $(t_c, t_{c+1}]$. Figure 10.4 on page 427 illustrates many of the important relationships that were just discussed.

10.5.1 Important Commonalities

Vector V can be viewed as consisting of three subvectors of ranked documents. These subvectors are referred to as the *prefix to the document cut-off subvector*, the *document cut-off subvector*, and the *suffix to the document cut-off subvector*. The prefix and suffix are defined as

$$V_{\text{pre}} = <v_1, v_2, \ldots, v_{t_c}>$$

and

$$V_{\text{suf}} = <v_{t_{c+1}+1}, v_{t_{c+1}+2}, \ldots, v_N>,$$

respectively.

The documents in V_{suf} occur after the document cut-off k and, hence, cannot affect the rankings in the portion of V (i.e., $< v_1, v_2, \ldots, v_{t_{c+1}} >$) that all of the performance measures, that can calculate their values at an arbitrary document cut-off point k, are used to calculate their values. This observation allows the performance measure calculations to ignore the documents in all of the equivalence classes that come *after* E_c in the ranking.

The number of possible document sequences that correspond to equivalence classes E_1, E_2, ..., and E_c is

$$\prod_{i=1}^{c} \frac{n_i!}{r_i!(n_i - r_i!)} \tag{10.5.1}$$

because the documents in each equivalence class can be arranged independently of those in any other equivalence class and each class has, at most, two kinds of documents –

relevant and non-relevant ones. It is well-known in combinatorics that the number of distinct permutations of n documents where there are x of one kind, y of another kind, and $n = x + y$, is

$$n!/(x!y!) = \binom{n}{x} = \binom{n}{y}.$$

Therefore, Equation 10.5.1 on the preceding page can be written more succinctly as

$$\prod_{i=1}^{c} \binom{n_i}{r_i}.$$

For some of the performance measure derivations that follow this one, it is important to remember that for each of the

$$\prod_{i=1}^{c-1} \binom{n_i}{r_i}$$

possible distinct sequences in V_{pre}, there are

$$\binom{n_c}{r_c}$$

possible V_c sequences that are associated with it from the equivalence class E_c. Similarly, each of the

$$\binom{n_c}{r_c}$$

possible distinct sequences in V_c has

$$\prod_{i=1}^{c-1} \binom{n_i}{r_i}$$

possible distinct V_{pre} sequences associated with it. For computational purposes, the V_c sequences that correspond to those in V_{pre} can be viewed as a table of $\binom{n_c}{r_c}$ sets of V_c sequences. Likewise, the V_{pre} sequences that correspond to those in V_c can be viewed as a table of

$$\prod_{i=1}^{c-1} \binom{n_i}{r_i}$$

sets of V_{pre} sequences. This means that the combined number of sequences contained in each table is

$$\prod_{i=1}^{c} \binom{n_i}{r_i}.$$

Figure 10.5 depicts these relationships.

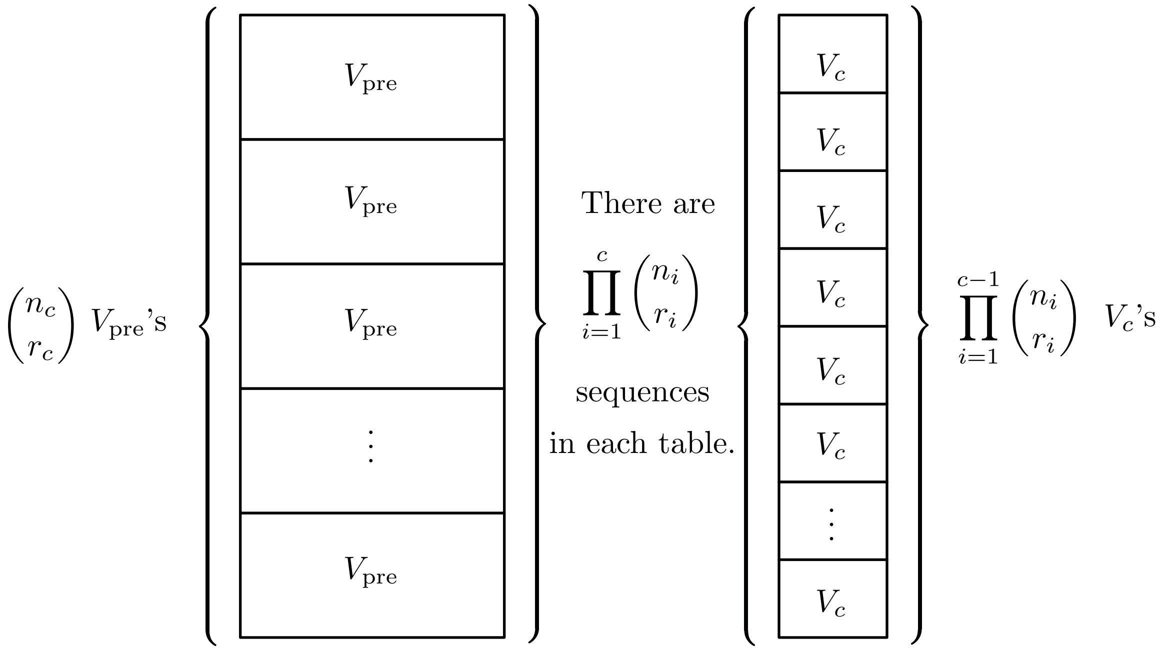


Figure 10.5: Each of the $\prod_{i=1}^{c-1} \binom{n_i}{r_i}$ distinct sequences in V_{pre} has $\binom{n_c}{r_c}$ distinct sequences in V_c that are associated with it. Conversely, each of the $\binom{n_c}{r_c}$ distinct sequences in V_c has $\prod_{i=1}^{c-1} \binom{n_i}{r_i}$ distinct V_{pre} sequences associated with it. This means that the combined number of sequences contained in each table in this diagram is $\prod_{i=1}^{c} \binom{n_i}{r_i}$.

10.5.2 Commonalities for Precision, Recall, and Average Search Length

The precision and recall measures are often defined with respect to the first $1 \leq k \leq N$ documents of a ranked vector V of documents for a query q. By contrast, the ASL measure is defined on the totality of a ranking. The definitions commonly encountered in the information retrieval (IR) literature for these measures assume that the ranking is strongly ordered.

The reason that these measures are being jointly treated as a group in this subsection is because the derivation of the combinatoric-based Type-T equations for them, at document cut-off k, have certain commonalities that make it easier to treat them together. The equations for precision and recall that are derived later in Section 10.6 are used later in that section to derive the equation for the MZE measure. Later, we show that deriving the document cut-off and Type-T version of the equation for the MZE measure is trivial once we have the corresponding equations for the precision and recall measures.

The Commonality That is Present in E_c

The commonality is the derivation of an equation for counting the number of relevant documents among those documents in the cut-off window of $k - t_c$ documents over all of the possible sequences of n_c documents in V_c. Each V_c sequence has r_c relevant documents and $n_c - r_c$ non-relevant documents. The document cut-off window for any sequence that is a member of V_c always consists of the first $k - t_c$ documents in that sequence.

The notion of document cut-off, as was originally explained, was based on a fixed position $1 \leq k \leq N$ in a vector V of ranked documents. This position is independent of the characteristics of any performance measure. It is just an arbitrary value that is decided upon prior to the ranking of a collection of documents for a query. In TREC-1 (Voorhees and Harman, 2005), document cut-offs of 5, 15, 30, 100, and 200 were used to restrict the calculation of the various performance measure values to a relatively small proportion of the documents that were at the front of the rankings. For example, a document cut-off of 30 means that the performance measure calculations only consider the first 30 documents in a ranking. All later documents in such a ranking, no matter how many, are ignored.

The notion of document cut-off is more nuanced than was explained in the previous paragraph. Depending on query and collection characteristics, the *effective document cut-off equivalence class* may be an equivalence class E_a that precedes equivalence class E_c. For example, the RR measure is a function of both the location in the ranking where the first relevant document occurs and of the document cut-off k. This document may occur in an equivalence class E_a that precedes the one (i.e., E_c) that is associated with document cut-off k. If this is the case, the documents in E_c can be ignored because they have no effect on the calculation of the reciprocal rank measure. Actually, a stronger statement can be made: the documents in *all* the equivalence classes that succeed those in E_a can be ignored when calculating the Type-T version of the RR measure at document cut-off k when the first relevant document occurs in an equivalence class that precedes E_c. This is discussed further in Section 10.6.6 (Reciprocal Rank).

Looking ahead, we find that the document cut-off equivalence class for the ASL, E, precision, and recall measures are totally determined by the value of k. The effective document cut-off equivalence class for these measures coincides with the one that is determined by k. However, the situation is different for the ESL and reciprocal rank measures. The effective document cut-off equivalence class for the ESL measure is a function of the number of requested documents *and* the document cut-off value k. The document cut-off equivalence class for the RR measure is a function of where the first relevant document occurs at *and* the document cut-off value k. Therefore, the effective document cut-off classes for these latter two measures may differ from the one that contains the kth document.

A More General Notion of Document Cut-off

In order to make the discussions in the remaining sections of this chapter more understandable, and, also, to simplify some of the calculations, we define a more general version of E_c. Previously, E_c was defined in terms of a fixed position k from the front of a vector V of ranked documents. This definition was adequate for the measures (e.g., ASL, E, precision, recall) where the equivalence class E_c for document cut-off k was independent of the query and document collection combination. On the contrary, this definition was inadequate for the measures (e.g., ESL and RR) where the effective document cut-off could occur in an *effective document cut-off equivalence class* $E_{\tilde{c}}$ that preceded E_c. Our revised notion of document cut-off is defined to take into account that the effective document cut-off equivalence class may not be solely dependent on the value of k and may occur in an equivalence class that is indexed by $\tilde{c}$, where $\tilde{c} < c$.

We propose the following three additional equivalence classes for vector V: $E_{c_{\text{fr}}}$ (the equivalence class that contains the first relevant document), $E_{c_{\text{xr}}}$ (the equivalence class that contains the xth relevant document), and $E_{c_{\text{k}}}$ (the equivalence class that contains the kth document). The $E_{c_{\text{fr}}}$ equivalence class is only applicable to the reciprocal rank derivations. The $E_{c_{\text{xr}}}$ equivalence class is only applicable to the expected search length derivations. The $E_{c_{\text{k}}}$ equivalence class is applicable to all of the derivations.

For a particular collection and query combination, equivalence class $E_{\tilde{c}}$ is exactly one of $E_{c_{\text{fr}}}$, $E_{c_{\text{xr}}}$, or $E_{c_{\text{k}}}$; that is, the value of the index $\tilde{c}$ is a value from the set $\{c_{\text{fr}}, c_{\text{k}}, c_{\text{xr}}\}$. The determination of the value of $\tilde{c}$ is specific to a performance ranking measure and is discussed in the upcoming sections of this chapter. The document cut-off value that is used in the calculations of the ESL and RR measures may be different than the one which was originally specified for the length of the document cut-off window; this can occur when $\tilde{c} \neq c_{\mathrm{k}}$.

In order to handle these situations, we introduce the notion of an *effective document cut-off* $\tilde{k}$ because, in the derivations for the ESL and reciprocal rank measures, it is necessary to differentiate between the specified document cut-off k and the effective document cut-off $\tilde{k}$. The value of $\tilde{k}$ is never any greater than that of k because an effective cut-off value, by definition, can never point to any document that occurs in an equivalence class that succeeds $E_{c_{\mathrm{k}}}$. The values of k and $\tilde{k}$ are identical for the ASL, recall, precision, and MZE measures. Similarly, the values of c and $\tilde{c}$ are also identical for these measures. This gives us

$$\tilde{k} = k,$$

$$\tilde{c} = c = c_{\mathrm{k}}, \text{ and}$$

$$E_{\tilde{c}} = E_c = E_{c_{\mathrm{k}}}.$$

When discussing the derivations for the ASL, recall, precision, and MZE measures, E_c, instead of $E_{\tilde{c}}$, and k, instead of $\tilde{k}$, is used for simplicity. The derivations for the ESL and RR measures uses $E_{\tilde{c}}$ and $\tilde{k}$ because the effective cut-off value $\tilde{k}$ for these measures may be less than the specified cut-off value k.

An Equation for the Number of Relevant Documents at Cut-off k

The information in Figure 10.6 on the following page can be used to derive an equation for the number of relevant documents in the document cut-off window over all of the possible sequences that could occupy that window of $k - t_c$ documents for the ASL, precision, recall, and MZE performance measures.

The minimum number of relevant documents m could be as few as 0. The maximum number could be as many as the minimum of the size of the window (i.e., $k - t_c$) and the number of relevant documents (i.e., r_c) in equivalence class E_c. This means that the

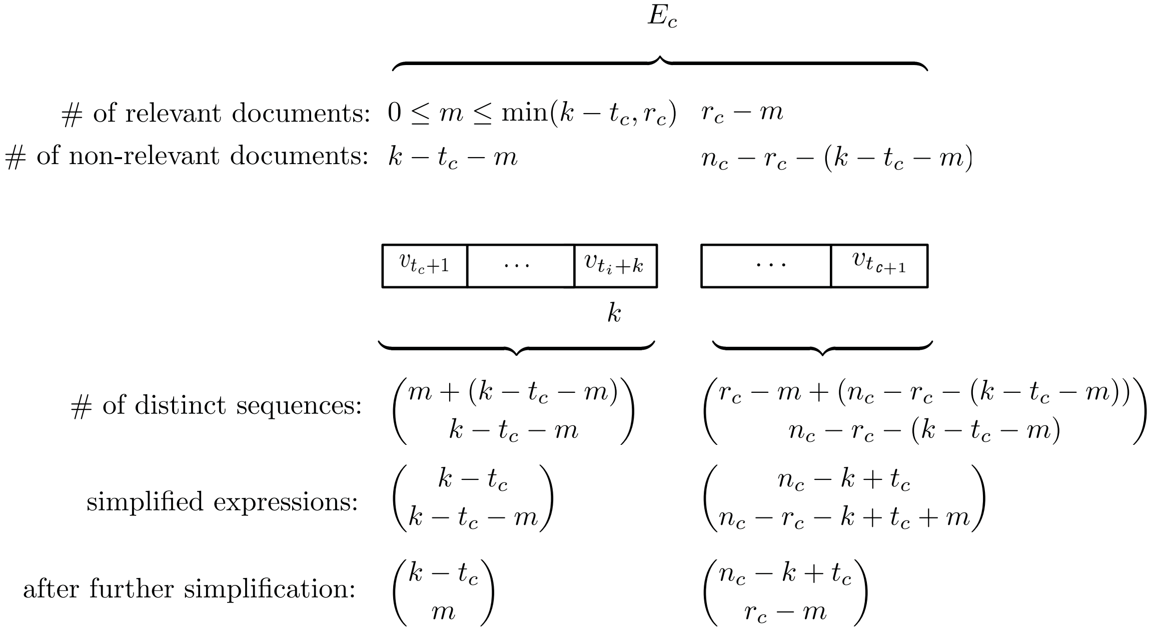

E_c
# of relevant documents: $0 \le m \le \min(k - t_c, r_c)$ $r_c - m$
# of non-relevant documents: $k - t_c - m$ $n_c - r_c - (k - t_c - m)$
v_{t_c+1} $\cdots$ v_{t_i+k} $\cdots$ $v_{t_{c+1}}$
k
# of distinct sequences: $\binom{m + (k - t_c - m)}{k - t_c - m}$ $\binom{r_c - m + (n_c - r_c - (k - t_c - m))}{n_c - r_c - (k - t_c - m)}$
simplified expressions: $\binom{k - t_c}{k - t_c - m}$ $\binom{n_c - k + t_c}{n_c - r_c - k + t_c + m}$
after further simplification: $\binom{k - t_c}{m}$ $\binom{n_c - k + t_c}{r_c - m}$

Figure 10.6: This diagram details the basic relationships that are associated with the equivalence class E_c for the ASL, precision, recall, and MZE measures. The equivalence class E_c contains document cut-off k for each of these measures.

expression $0 \le m \le \min(k - t_c, r_c)$ describes the relationship that must hold for the number of relevant documents that could be in the document window for V_c. The corresponding number of non-relevant documents that are in this window can be calculated by subtracting the value of m from the size of the window. This yields the expression $k - t_c - m$ for the number of non-relevant documents that can occupy the remaining positions in this window. From these expressions, we can state that the number of distinct ways that the documents could be arranged in the document cut-off window is

$$\binom{k - t_c}{k - t_c - m} = \binom{k - t_c}{m}.$$

Now, we need to develop analogous expressions for the number of distinct ways that the remaining

$$(r_c - m) + (n_c - r_c - (k - t_c - m)) = n_c - (k - t_c)$$

documents can be arranged in the part of V_c that comes after the document cut-off window. The $r_c - m$ expression in the above equation represents the number of relevant documents that occupy some of the remaining positions in V_c and the

$$n_c - r_c - (k - t_c - m)$$

expression in this equation represents the number of non-relevant documents are placed in these remaining positions. From this information, we can state that the number of distinct ways that the remaining $n_c - (k - t_c)$ documents in V_c can be arranged is

$$\binom{n_c - (k - t_c)}{r_c - m}.$$

Since these two sets of documents can be arranged independently of each other, the expression

$$\binom{k-t_c}{m}\binom{n_c-(k-t_c)}{r_c-m}$$

represents the number of distinct sequences of documents for V_c that can occupy the document cut-off window, when that window contains exactly m relevant documents. The expression

$$\sum_{m=0}^{\min(k-t_c,r_c)}\binom{k-t_c}{m}\binom{n_c-(k-t_c)}{r_c-m}$$

represents the number of distinct document sequences for V_c, over all of the possible numbers of relevant documents that can occupy positions in the document cut-off window.

Our next goal is to derive a closed form expression for the immediately previous expression. First, notice that the immediately previous expression can be simplified to

$$\sum_{m}\binom{k-t_c}{m}\binom{n_c-(k-t_c)}{r_c-m}$$

because the term $\binom{k-t_c}{m}$ vanishes when either m is negative or $m > k-t_c$, and the term $\binom{n_c-(k-t_c)}{r_c-m}$ vanishes when $m > r_c$. Finally, we can apply Equation 10.4.8 on page 423, one of the convolution identities, to obtain

$$\binom{n_c}{r_c},$$

the closed form version of the expression. This expression is used in the derivation of the equations for the ASL, precision, and recall measures.

The sum of the ranks of the relevant documents in the document cut-off window, over all possible sequences of documents in V_c, is also helpful for determining later derivations. More specifically, this is something that is necessary to know for the derivation of the ASL. It is not necessary, nor used, for the precision and recall derivations. Of course, it is possible, perhaps even desirable, that the author of this dissertation should wait until the ASL derivation section to do this derivation. However, the author chose to do it here because, from the discussion in the previous paragraphs of this subsection, both the reader and the author are already familiar with how to develop the counts for E_c.

The expression for the sum of the ranks can be obtained by weighting the summand value in

$$\sum_m \binom{k-t_c}{m}\binom{n_c-(k-t_c)}{r_c-m}$$

by m, the value of the index variable at the time the summand is evaluated. This gives us

$$\begin{aligned}
\sum_m m\binom{k-t_c}{m}\binom{n_c-(k-t_c)}{r_c-m} &= (k-t_c)\sum_m \binom{k-t_c-1}{m-1}\binom{n_c-(k-t_c)}{r_c-m} && (10.5.2)\\
&= (k-t_c)\sum_{m+1\geq 0} \binom{k-t_c-1}{m}\binom{n_c-(k-t_c)}{r_c-1-m} && (10.5.3)\\
&= (k-t_c)\sum_{m\geq -1} \binom{k-t_c-1}{m}\binom{n_c-(k-t_c)}{r_c-1-m} && (10.5.4)\\
&= (k-t_c)\sum_m \binom{k-t_c-1}{m}\binom{n_c-(k-t_c)}{r_c-1-m} && (10.5.5)\\
&= (k-t_c)\binom{n_c-1}{r_c-1}. && (10.5.6)
\end{aligned}$$

This value is the sum of the ranks for the relevant documents in the document cut-off window over all the possible distinct sequences that can appear in V_c.

Before continuing this discussion, it would be useful to explain some parts of the derivation of the previous equation. Equation 10.5.3 was produced from Equation 10.5.2 by doing a change of variable (i.e., replace m by $m+1$). This starts a series of operations that help to simplify the equation. A simplification of the summation limits occurs at Equation 10.5.4. Further simplification of the limits occur at Equation 10.5.5 because the first binomial of the product that is being summed vanishes when m is negative. Finally, we go from Equation 10.5.5 to Equation 10.5.6 by applying either Equation 10.4.8 or Equation 10.4.9.

The proportion

$$\binom{n_c - 1}{r_c - 1} / \binom{n_c}{r_c}$$

is useful later in the derivations of the expressions for precision and recall. We can simplify it in this way:

$$\begin{aligned}
\binom{n_c - 1}{r_c - 1} / \binom{n_c}{r_c} &= \frac{(n_c - 1)!}{(r_c - 1)!(n_c - r_c)!} \left(\frac{n_c!}{r_c!(n_c - r_c)!} \right)^{-1} && (10.5.7) \\
&= \frac{(n_c - 1)!}{(r_c - 1)!(n_c - r_c)!} \frac{r_c!(n_c - r_c)!}{n_c!} && (10.5.8) \\
&= \frac{(n_c - 1)!}{(r_c - 1)!(n_c - r_c)!} \frac{r_c(r_c - 1)!(n_c - r_c)!}{n_c(n_c - 1)!} && (10.5.9) \\
&= \frac{r_c}{n_c}. && (10.5.10)
\end{aligned}$$

Before continuing this discussion, it would be useful to explain some parts of the derivation of the previous equation. Equation 10.5.8 was produced from Equation 10.5.7 by re-expressing the negative power term as a reciprocal. The simplification continued by expanding the terms $r_c!$ and $n_c!$ in Equation 10.5.8 into, respectively, the terms $r_c(r_c - 1)!$ and $n_c(n_c - 1)!$ in Equation 10.5.9. Finally, there was a transition from Equation 10.5.9 to Equation 10.5.10 that was effected by canceling all the terms in the numerator that also appeared in the denominator.

An expression for the the mean number of relevant documents in the document cut-off window of V_c, over all of is possible sequences of ranked documents, can be obtained by dividing the expression

$$(k - t_c) \binom{n_c - 1}{r_c - 1},$$

from Equation 10.5.6 on the preceding page, by $\binom{n_c}{r_c}$, the number of possible sequences of ranked documents for E_c. The resultant expression is

$$(k - t_c)\binom{n_c - 1}{r_c - 1}\binom{n_c}{r_c}^{-1}.$$

By the result of Equation 10.5.10 on the previous page, this expression can be simplified to

$$(k - t_c)\frac{r_c}{n_c} = \frac{(k - t_c)r_c}{n_c}.$$

10.6 Derivations for the ESL, ASL, Precision, Recall, MZE, and RR Measures

The next 6 subsections discuss the derivation of the equations for the ESL, ASL, precision, recall, MZE, and RR performance measures. They also discuss the derivations of the Type-T versions of the precision and recall measures because the Type-T version of the MZE measure is defined in terms of these two measures. We start our discussion with the derivation for the ESL measure. The main reason for this is historical because the ESL measure appears to be the first performance measure in the IR literature that correctly calculates performance for rankings that may be weakly-ordered.

10.6.1 Expected Search Length

Earlier in this chapter, we mentioned that the ESL can be viewed as being both defined on the totality of a ranking and as a point measure. The ESL measure assumes that the ranking may be weakly-ordered and is defined as

$$\text{ESL}(V, x) = \begin{cases} 0, & \text{if } x < 1; \\ j + s\dfrac{i}{r+1}, & \text{if } 1 \le x \le R; \text{ and} \\ |V|, & \text{otherwise;} \end{cases} \tag{10.6.1}$$

where V is a ranked vector of documents for a query q, R is the total number of relevant documents in the collection, l is the level at which the x^{th} relevant document occurs, j is the total number of documents not relevant to q in all levels which precede level l in the weak ordering, i is the number of documents not relevant to q in level l, s is the number such that the s^{th} relevant document found in level l of the weak ordering would complete the search for request q, and r is the number of documents level l which are relevant to q.

There are two differences in the above definition from that given in Cooper (1968). The ESL measure, as defined above (shown by Equation 10.6.1), has been extended to handle the situation where the requested number of relevant documents is less than 1 and to handle the situation where the requested number of relevant documents is larger than the available number of relevant documents. When the requested number of relevant documents is less than 1, the value of the ESL is 0 because no documents have to be examined. When the requested number of relevant documents is greater than N, Kraft and Lee (1979) defines the expected value to be the same as the number of documents in the collection because the entire collection has to be examined in order to determine that there are an insufficient number of relevant documents to satisfy the request for x relevant documents.

We proceed by deriving the analogous set of equations for ESL@$k(V, x)$, the ESL at document cut-off k. We also show that the Cooper equation (the middle one in Equation 10.6.1) is a special case of the more general equation for ESL@$k(V, x)$. Cooper (1968) states this middle equation but did not provide its derivation in the article. He did say that, upon request, a mathematical supplement (Cooper, 1967) to his article on the ESL was "[o]btainable by mail from the Graduate Library School, University of Chicago, Chicago, Ill." The author of this dissertation was not successful in obtaining a copy of this supplement. That is another reason for providing this derivation and for showing that it is equivalent to the equation by Cooper. Other reasons for providing this derivation include showing how these equations can be derived by combinatoric techniques; illustrating arguments that are similar to those that are used for the document cut-off k versions of the ASL, MZE, and RR measures; and operationalizing the equation in terms of the tie-breaking framework and its notation.

The Derivation for ESL@$k(V, x)$

The number of non-relevant documents contained in any sequence of V_{pre} is the same as the number that is contained in any other of these sequences. This is true because the various document permutations have no effect on the number of non-relevant documents that are contained by each sequence. That number is the same no matter what positions the non-relevant documents occupy. Following Cooper's notation, this value is denoted by the symbol j. It can be defined as

$$j = N_{c_{\text{xr}}} - R_{c_{\text{xr}}},$$

where $N_{c_{\text{xr}}}$ and $R_{c_{\text{xr}}}$ denote, respectively, the number of documents and the number of relevant documents that are contained in the equivalence classes that precede equivalence class $E_{c_{\text{xr}}}$. Note that this equivalence class may be different than the one (i.e., $E_{c_{\text{k}}}$) that the document cut-off k is associated with. The discussion to follow assumes that the xth relevant document occurs either in subvector $V_{c_{\text{k}}}$ or in a subvector that precedes it. The case where the xth relevant document would occur in a subvector that succeeds $V_{c_{\text{k}}}$ is handled separately because it corresponds to a situation where the requested number of relevant documents do not exist among the first k documents of any of the sequences of V that are comprised of documents in the first c_{k} equivalence classes.

For a vector V, the values for c_{k} and c_{xr} are defined as follows. Let c_{k} denote the index of the equivalence class that contains the kth rank. Its definition is

$$c_{\mathrm{k}} = \begin{cases} \min(S), & \text{if } \operatorname{card}(S) > 0; \\ \text{undefined}, & \text{otherwise}; \end{cases}$$

where

$$S = \left\{ i \,\middle|\, \left(\left(\sum_{1 \le p \le i} n_p \right) \ge k \right) \text{ and } 1 \le i \le m \right\}$$

and m is the number of equivalence classes that are associated with vector V. From earlier discussions, we know that the number of equivalence classes that are associated with a vector V is the same as the number of distinct RSVs that are associated with V. Let c_{xr} denote the index of the equivalence class that contains the xth relevant document. Its definition is

$$c_{\mathrm{xr}} = \begin{cases} \min(S), & \text{if } \operatorname{card}(S) > 0; \\ \text{undefined}, & \text{otherwise}; \end{cases}$$

where

$$S = \left\{ i \,\middle|\, \left(\left(\sum_{1 \le p \le i} r_p \right) \ge x \right) \text{ and } 1 \le i \le m \right\}$$

and m is the number of equivalence classes that are associated with vector V.

Next, we need to calculate the mean number of non-relevant documents that can appear before the sth relevant document when the specified cut-off value is k. The effective cut-off position and cut-off equivalence class index pair $(\tilde{k}, \tilde{c})$ is determined as follows:

$$(\tilde{k}, \tilde{c}) = \begin{cases} (k, c_{\mathrm{k}}), & \text{if } c_{\mathrm{xr}} = c_{\mathrm{k}}; \\ (t_{c_{\mathrm{xr}}+1}, c_{\mathrm{xr}}), & \text{if } c_{\mathrm{xr}} < c_{\mathrm{k}}. \end{cases}$$

If $c_{\mathrm{xr}} > c_{\mathrm{k}}$, then the relevant document that satisfies the request is in an equivalence class that comes after $E_{c_{\mathrm{k}}}$ (the equivalence class that contains the kth document). In this case, the request cannot be satisfied and, consequently, $\mathrm{ESL@k(V,x)} = k$.

In the other two cases, where $c_{\mathrm{xr}} \leq c_{\mathrm{k}}$, the ESL@$k(V, x)$ value can be calculated by determining several quantities. These quantities are referenced several times in the set of expressions that are developed below. Let

$$\begin{aligned} X &= \sum_{p=1}^{\tilde{c}-1} (n_p - r_p) \\ &= N_{\tilde{c}} - R_{\tilde{c}} \end{aligned}$$

denote the number of non-relevant documents that precede the documents in $V_{\tilde{c}}$ and let

$$Y = \binom{n_{\tilde{c}}}{r_{\tilde{c}}}$$

denote the number of sequences that are in $V_{\tilde{c}}$.

The summation limit, $\min(i, \tilde{k} - t_{\tilde{c}} - s)$, represents the number of non-relevant documents that can be mixed in with the $s - 1$ relevant documents that precede the sth relevant one. With our assumption that the sth relevant document occurs at, or before, point k in a ranking, the capacity of the cut-off window is $\tilde{k} - t_{\tilde{c}}$. At least s of these slots are occupied by relevant documents. The most non-relevant documents that can precede the sth relevant document is the minimum of the total number of relevant documents i and the capacity $\tilde{k}$ of the effective cut-off window minus the slots for the s relevant documents.

From information that was obtained from Figure 10.7 on page 448, we see that the $\binom{(s-1)+m}{s-1}$ term in Equation 10.6.2 represents the number of ways that $s-1$ relevant documents and m non-relevant ones can be ranked ahead of the sth relevant document when the summation limit is the minimum of the values for i (the number of non-relevant documents in $V_{\tilde{c}}$) and $\tilde{k} - t_{\tilde{c}} - s$ (the maximum number of remaining documents that can occupy slots in the document cut-off window once s relevant documents have been selected for that window). The $\binom{(r-s)+(i-m)}{r-s}$ term represents the number of ways that $r-s$ relevant documents and $i-m$ non-relevant ones can be ranked behind the sth relevant one. Since the two sets of orderings are independent, the total number of orderings (T_m) is simply the product of the two independent orderings. That is, we have

$$T_m = \binom{(s-1)+m}{s-1}\binom{(r-s)+(i-m)}{r-s}. \qquad (10.6.2)$$

This means that the total number of non-relevant documents that are ranked ahead of the sth relevant document for a particular value of m is mT_m. Summing these up for all values of m in the range $[0, \min(i, \tilde{k} - t_{\tilde{c}} - s)]$ results in these next two equations.

$$\begin{aligned} A &= mT_m \\ &= \sum_{m=0}^{\min(i,\tilde{k}-t_{\tilde{c}}-s)} m\binom{(s-1)+m}{s-1}\binom{(r-s)+(i-m)}{r-s} \end{aligned} \qquad (10.6.3)$$

and

$$\begin{aligned} B &= T_m, \\ &= \sum_{m=0}^{\min(i,\tilde{k}-t_{\tilde{c}}-s)} \binom{(s-1)+m}{s-1}\binom{(r-s)+(i-m)}{r-s}, \end{aligned} \qquad (10.6.4)$$

where the variable A denotes the number of non-relevant documents that can appear before the sth relevant document and the variable B denotes the number of sequences in which the non-relevant documents can appear before the sth relevant document.

From our experience with the manipulation of combinatoric identities, we notice that Equation 10.6.3 on the preceding page seems to be a possible candidate for simplification due to the expression

$$m\binom{(s-1)+m}{s-1}$$

that appears in it. By algebraic and combinatorial manipulations, this expression can be simplified in this way:

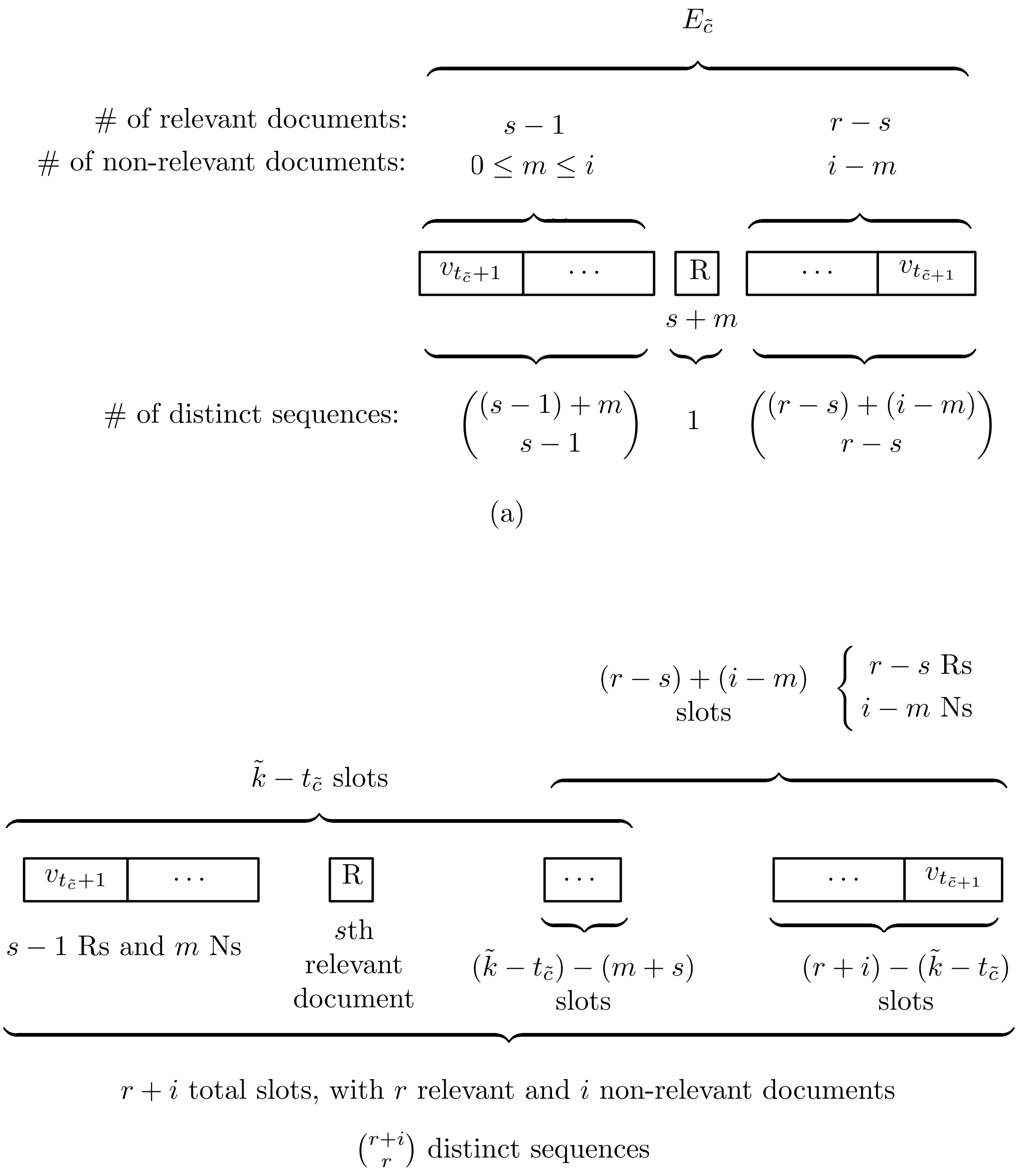
$$\begin{aligned}
m\binom{(s-1)+m}{s-1} &= m\binom{(s-1)+m}{m} && \text{(by Equation 10.4.6 on page 422)}\\
&= m\frac{((s-1)+m)!}{m!(s-1)!} && \text{(by Equation 10.4.5 on page 422)}\\
&= \frac{((s-1)+m)!}{(m-1)!(s-1)!} && \text{(by dividing numerator and denominator by } m\text{)}\\
&= s\frac{((s-1)+m)!}{(m-1)!s!} && \text{(by multiplying numerator and denominator by } s\text{)}\\
&= s\binom{(s-1)+m}{m-1} && \text{(by Equation 10.4.5 on page 422)}\\
&= s\binom{(s-1)+m}{s}. && \text{(by Equation 10.4.6 on page 422)}
\end{aligned}$$

Therefore, Equation 10.6.3 on the previous page can be rewritten as

$$A = s\sum_{m=0}^{\min(i,\tilde{k}-t_{\tilde{c}}-s)} \binom{(s-1)+m}{s}\binom{(r-s)+(i-m)}{r-s}. \tag{10.6.5}$$

If we let

$$\begin{aligned}
C &= \binom{n_{\tilde{c}}}{r_{\tilde{c}}} - \sum_{m=0}^{\min(i,\tilde{k}-t_{\tilde{c}}-s)} \binom{(s-1)+m}{s-1}\binom{(r-s)+(i-m)}{r-s}\\
&= Y - B
\end{aligned}$$



(a)

(b)

Figure 10.7: This diagram details the relationships that are associated with the equivalence class $E_{\tilde{c}}$ for the ESL measure. Equivalence class $E_{\tilde{c}}$ contains a total of $r+i$ documents. It has r relevant documents and i non-relevant ones. The variable s denotes the number of relevant documents that would complete the request for a query q. The box with a relevant document R inside it represents that the sth relevant document occurs at position $s+m$ in subvector $V_{\tilde{c}}$. The variable m denotes the number of non-relevant documents that appear before the sth relevant document.

denote the number of sequences in which it is impossible for x requested relevant documents to appear at, or before, document cut-off k in a sequence; then, with these equations that we developed in the last several paragraphs, we can write the equation for the ESL at position k in a ranked vector V of documents for a request of x documents as

$$\begin{aligned}
\text{ESL@}k(V,x) &= \frac{XB + A + kC}{Y} \\
&= \frac{(N_{\tilde{c}} - R_{\tilde{c}})B + A + k(Y - B)}{Y} \\
&= \frac{N_{\tilde{c}}B - R_{\tilde{c}}B + A + kY - kB}{Y} \\
&= \frac{N_{\tilde{c}}B - R_{\tilde{c}}B + A - kB}{Y} + \frac{kY}{Y} \\
&= k + \frac{B(N_{\tilde{c}} - R_{\tilde{c}} - k) + A}{Y}. \qquad (10.6.6)
\end{aligned}$$

The Complete Equation for ESL@$k(V,x)$

The derivations that resulted in Equation 10.6.6 assumed that at least one relevant document was requested, but that the total number being requested did not exceed the maximum number M_x of relevant documents that was possible in vector V between positions 1 and $\tilde{k}$, inclusive. This maximum value, for a subvector $V_{\tilde{c}}$, is determined by first totaling the number of relevant documents that appear in the subvectors that precede subvector $V_{\tilde{c}}$. More formally, these are the subvectors V_i, where $1 \leq i < \tilde{c}$.

From previous discussions, we know that $R_{\tilde{c}}$ denotes the total number of relevant documents in these subvectors. The final step in determining the value of M_x is counting the maximum number of relevant documents that are possible in the first $\tilde{k}-t_{\tilde{c}}$ positions of subvector $V_{\tilde{c}}$. This number cannot exceed $\tilde{k}-t_{\tilde{c}}$ (the number of positions in the document cut-off window), nor can it exceed $r_{\tilde{c}}$ (the number of relevant documents in subvector $V_{\tilde{c}}$). From these two constraints, we can surmise that the maximum possible number of relevant documents that can appear in the document cut-off window is the minimum of these two values, that is, $\min(\tilde{k} - t_{\tilde{c}}, r_{\tilde{c}})$. If we combine this information with our prior information, we can state that

$$M_x = R_{\tilde{c}} + \min(\tilde{k} - t_{\tilde{c}}, r_{\tilde{c}}).$$

Therefore, Equation 10.6.6 on the previous page is valid when $1 \leq x \leq (R_{\tilde{c}} + \min(\tilde{k} - t_{\tilde{c}}, r_{\tilde{c}}))$. When x (the number of requested relevant documents) is zero, or less, the ESL is 0 because there are no relevant documents to obtain for this value of x. When the requested number of relevant documents exceed the number that are possible in the positions that start at the beginning of the vector V and end at the last position in the document cut-off window, the ESL is the document cut-off value k. The discussions in this subsection allow us to state that the complete equation for ESL@$k(V, x)$ is

$$\text{ESL@}k(V, x) = \begin{cases} 0, & \text{if } x \leq 0; \\ k, & \text{if } x > (R_{\tilde{c}} + \min(\tilde{k} - t_{\tilde{c}}, r_{\tilde{c}})); \\ k + \frac{B(N_{\tilde{c}} - R_{\tilde{c}} - k) + A}{Y}, & \text{otherwise.} \end{cases} \tag{10.6.7}$$

Cooper's Equation as a Special Case of ESL@$k(V, x)$

Basically, the Cooper equation corresponds to the situation where *all* of the non-relevant documents in the collection of N documents can appear before the sth relevant document. In other words, the size of the document cut-off window is $k = N$, which means that, effectively, the performance measure is based on all of the documents instead of just the first $k < N$ documents in a ranking. This implies that $\min(i, \tilde{k} - t_{\tilde{c}} - s) = i$.

If we use A to denote the number of non-relevant documents that can appear before the sth relevant document and use B to denote the number of sequences in which the non-relevant documents can appear before the sth relevant document, we have

$$\text{ESL@}k(V, x) = j + A/B. \tag{10.6.8}$$

A consequence of the values for i and $\min(i, \tilde{k} - t_{\tilde{c}} - s)$ being equal is that the equations for both A and B can be simplified to closed forms by making use of a well-known convolution identity. Equation 10.4.10 on page 424 states that

$$\binom{l+q+1}{m+n+1} = \sum_{k=0}^{l} \binom{l-k}{m}\binom{q+k}{n},$$

where the conditions $n \geq 0$ and $0 \leq m \leq l$ must be true in order for the application of this identity to be valid.

In order to minimize confusion with the symbols that appear in Equation 10.6.3 on page 446 and Equation 10.6.4 on page 446, we use the following equivalent identity by substituting the dummy variables a, b, c, d, and e for the dummy variables k, l, m, n, and q, respectively. These substitutions yield

$$\binom{b+e+1}{c+d+1} = \sum_{a=0}^{b} \binom{b-a}{c}\binom{e+a}{d},$$

where the conditions $d \geq 0$ and $0 \leq c \leq b$ must be true in order for the application of this identity to be valid.

Based on the following symbol correspondence between it and Equation 10.6.5 on page 447, we have this mapping:

$$s - 1 \Longleftrightarrow e$$

$$r - s + i \Longleftrightarrow b$$

$$r - s \Longleftrightarrow c$$

$$m \Longleftrightarrow a$$
$$s \Longleftrightarrow d.$$

Before we can proceed, there are two questions that we need to ask and get affirmative answers to. We need to know if s is greater than zero (i.e., the value of s is a positive integer). The answer is yes because the minimum number of relevant documents that can be requested is 0. We also need to know if the relationship $0 \leq r - s \leq r - s + i$ is true. The answer to this question is also yes because we know that the number of non-relevant documents can never be negative. Therefore, it is valid to apply Equation 10.4.10 on page 424.

After using the mapping to make the appropriate substitutions, and later commuting the terms of the binomial product, we obtain

$$\begin{aligned}
A &= s \sum_{m=0}^{r-s+i} \binom{(s-1)+m}{s} \binom{(r-s)+(i-m)}{r-s} \\
&= s \sum_{m=0}^{r-s+i} \binom{(r-s)+(i-m)}{r-s} \binom{(s-1)+m}{s} && \text{(algebraic commutativity)} \\
&= s \sum_{m} \binom{(r-s)+(i-m)}{r-s} \binom{(s-1)+m}{s} && \text{(index simplification)} \\
&= s \binom{r+i}{r+1}. && \text{(by Equation 10.4.10 on page 424)}
\end{aligned}$$

Notice that, between the second and third steps of the derivation, the summation was simplified by replacing the lower and upper bounds on the index of summation with an unconstrained index that ranges over the entire set of integers. This transformation is valid because the first term of the binomial product vanishes when $m > i$ and the second term vanishes when $m \leq 0$. By the use of this same identity, we can also simplify B by assuming that the user always requests at least one relevant document.

$$\begin{aligned}
B &= \sum_{m=0}^{r-s+i} \binom{(s-1)+m}{s-1}\binom{(r-s)+(i-m)}{r-s} \\
&= \sum_{m} \binom{(s-1)+m}{s-1}\binom{(r-s)+(i-m)}{r-s} && \text{(index simplification)} \\
&= \binom{r+i}{r}. && \text{(by Equation 10.4.10 on page 424)}
\end{aligned}$$

The simplified expressions above for A and B allow the rewriting of Equation 10.6.8 on page 451 as

$$\begin{aligned}
\mathrm{ESL} &= j + A/B \\
&= j + s\frac{\binom{r+i}{r+1}}{\binom{r+i}{r}}. \qquad (10.6.9)
\end{aligned}$$

The fraction in the second line of Equation 10.6.9 can be simplified in this manner:

$$\begin{aligned}
\frac{\binom{r+i}{r+1}}{\binom{r+i}{r}} &= \frac{(r+i)!}{(r+1)!(i-1)!}\left(\frac{(r+i)!}{r!i!}\right)^{-1} \\
&= \frac{(r+i)!}{(r+1)!(i-1)!}\frac{r!i!}{(r+i)!} \\
&= \frac{r!i!}{(r+1)!(i-1)!} \\
&= \frac{i!}{(r+1)(i-1)!} \\
&= \frac{i}{r+1}.
\end{aligned}$$

After several combinatorial and algebraic manipulations, Equation 10.6.9 simplifies to

$$\mathrm{ESL} = j + s\frac{i}{r+1} \qquad (10.6.10)$$

where i is the number of non-relevant documents in $E_{c_{\mathrm{xr}}}$. If this simplified version is rewritten as

$$\mathrm{ESL} = j + \frac{is}{r+1}, \tag{10.6.11}$$

then it is identical to the equation that appeared in Cooper (1968).

10.6.2 Average Search Length

The Derivation for ASL@$k(V)$

The Type-D ASL measure is defined as

$$\mathrm{ASL}(V) = \begin{cases} \dfrac{\sum\limits_{i=1}^{N} i \cdot I_R(v_i)}{\sum\limits_{i=1}^{N} I_R(v_i)}, & \text{if } \left(\sum\limits_{i=1}^{N} I_R(v_i)\right) > 0;\ \text{and} \\ N+1, & \text{otherwise;} \end{cases}$$

where V is a ranked vector of N documents for a query q and I_R is an indicator function for a collection R of relevant documents. The value of $I_R(a)$ is equal to 1 if its argument a represents a relevant document and is equal to 0, otherwise. The effective cut-off position and cut-off equivalence class index pair $(\tilde{k}, \tilde{c})$ is equal to (k, c_{k}).

The Average Search Length at document cut-off k can be largely determined by totaling the ranks of the relevant documents, starting at rank 1 and including all ranks up to rank k, and then dividing that total by the number of relevant documents that were used to compute it. This approach works fine unless one or more of these k-length sequences do not have any relevant documents. Since all the documents in the collection are not necessarily being used, due to cut-off k, it is quite possible that, even though the collection has one or more relevant documents for a query q, none of them are guaranteed to appear in a ranking at or before rank k. Any ASL calculation that incorporates the notion of an arbitrary document cut-off k must take this case into account. A manner in which this can be handled is detailed below.

The derivation of the Type-T equation for the ASL proceeds in several steps. Ultimately, the resultant equation consists of an expression for a numerator n and a denominator d. The value of the ASL is calculated by the expression n/d. The value for the numerator n is the sum of the ranks, across all the sequences of length k, that have a relevant document associated with the rank *plus* the number of sequences that consist entirely of k non-relevant documents multiplied by the weight (i.e., $k+1$) for such a sequence. The denominator d is the number of the ranks, across all the sequences of length k, that have a relevant document associated with the rank *plus* the number of sequences that consist entirely of k non-relevant documents. The weight of a relevant document in the numerator is its rank; in the denominator, its weight is 1. The weight of a virtual document is $k+1$ in the numerator and 1 in the denominator. Virtual documents only enter the calculations for n and d when there are one or more k-length sequences of non-relevant documents, each with 1 as the starting rank of the respective sequence.

The value for n is the sum of three quantities, the same is true for the value for d. The expressions that represent the sub-expressions that help compute the values for these quantities are denoted by $A, B, C, D, E, F, G, X, Y$, and Z. Below, we discuss each of these in turn.

First, we define several quantities that are referenced several times in the set of expressions that is developed below. Let

$$X = \prod_{i=1}^{c-1} \binom{n_i}{r_i} \tag{10.6.12}$$

denote the number of sequences that are in V_{pre}, let

$$Y = \binom{n_c}{r_c}$$

denote the number of sequences that are in V_c, and let

$$Z = \frac{r_c}{n_c}\binom{n_c}{r_c} \tag{10.6.13}$$

denote the number of relevant documents that are in each column of V_c.

Now, let A denote the number of relevant documents in V_{pre}. Its value is the product of the number of sequences X in V_{pre} and the number of relevant documents R_c in each sequence of V_{pre}. The number of relevant documents is the same for each sequence in V_{pre} and can be calculated from the expression

$$R_c = \sum_{i=1}^{c-1} r_i.$$

Their combination yields

$$\begin{aligned} A &= \left(\prod_{i=1}^{c-1}\binom{n_i}{r_i}\right)\sum_{j=1}^{c-1} r_j \\ &= XR_c, \end{aligned}$$

the number of relevant documents in V_{pre}.

Let B denote the number of relevant documents that are in the size $k - t_c$ window of the $\binom{n_c}{t_c}$ sequences that are associated with V_c. These sequences can be visualized as a table where each sequence corresponds to a row that has n_c columns. The proportion of relevant documents in each column is r_c/n_c. The equation to calculate the number of relevant documents in this table is

$$\begin{aligned} B &= (k - t_c)\binom{n_c}{t_c}\frac{r_c}{n_c} \\ &= (k - t_c)Z. \end{aligned}$$

Let C denote the number of sequences in V_c that consist entirely of non-relevant documents in the document cut-off window. The expression to calculate its value is

$$C = \binom{n_c - (k - t_c)}{(n_c - r_c) - (k - t_c)}.$$

The top part of the binomial in this equation represents the number of documents that are not part of the window of size $k - t_c$ that consists entirely of non-relevant documents. This set of documents corresponds to those documents that lie outside this window and contains r_c relevant documents and $(n_c - r_c) - (k - t_c)$ non-relevant ones. The number of distinct sequences of length $n_c - (k - t_c)$ that can be constructed from these documents is represented by the binomial that is on the right hand side of the above equation.

Let D denote the sum of all the ranks in V_{pre} that are associated with a relevant document. Its value v can be partially calculated by determining the number of relevant documents in each column of V_{pre} (a value v that may vary according to the equivalence class, but, for an equivalence class E_i, is *constant* – that is, *each* of the n_i columns in V_{pre} has the *same* value v associated with it) and then multiplying that value by the rank for that column. When this is done for all the equivalence classes that precede E_c, and totaled, we obtain

$$\begin{aligned}
D &= \sum_{i=1}^{c-1} \sum_{j=t_i+1}^{t_{i+1}} \frac{r_i}{n_i} \left(\prod_{l=1}^{c-1} \binom{n_l}{r_l} \right) j \\
&= \left(\prod_{l=1}^{c-1} \binom{n_l}{r_l} \right) \sum_{i=1}^{c-1} \frac{r_i}{n_i} \sum_{j=t_i+1}^{t_{i+1}} j && \text{(by Equation 10.4.1 on page 421)} \\
&= \left(\prod_{l=1}^{c-1} \binom{n_l}{r_l} \right) \sum_{i=1}^{c-1} \frac{r_i}{n_i} \left[\frac{t_{i+1}(t_{i+1}+1)}{2} - \frac{t_i(t_i+1)}{2} \right] && \text{(by Equation 10.4.13 on page 424)} \\
&= \left(\prod_{l=1}^{c-1} \binom{n_l}{r_l} \right) \sum_{i=1}^{c-1} \frac{r_i}{n_i} \left[\binom{t_{i+1}+1}{2} - \binom{t_i+1}{2} \right] && \text{(by Equation 10.4.12 on page 424)} \\
&= X \left[\binom{t_{i+1}+1}{2} - \binom{t_i+1}{2} \right]. && \text{(by Equation 10.6.12 on page 455)}
\end{aligned}$$

The sub-expression

$$\frac{r_i}{n_i}\left(\prod_{l=1}^{c-1}\binom{n_l}{r_l}\right),$$

on the first line of the equation for D, represents the number of relevant documents that are associated with each column in V_{pre} for equivalence class E_i. The sub-expression

$$\sum_{j=t_i+1}^{t_{i+1}}\frac{r_i}{n_i}\left(\prod_{l=1}^{c-1}\binom{n_l}{r_l}\right)j,$$

where j denoted the weight for column j, represents the number of relevant documents in V_{pre} that are associated with equivalence class E_i (i.e., subvector V_i). Note that the weight for column j is simply its rank in vector V.

Let E denote the sum of the ranks of all the relevant documents in V_c. The computation of its value is similar to that for D. The expression to calculate its value is

$$\begin{aligned}
E &= \sum_{j=t_c+1}^{t_{c+1}}\frac{r_c}{n_c}\binom{n_c}{r_c}j \\
&= \frac{r_c}{n_c}\binom{n_c}{r_c}\sum_{j=t_c+1}^{t_k} j && \text{(by Equation 10.4.1 on page 421)} \\
&= \frac{r_c}{n_c}\binom{n_c}{r_c}\left[\frac{k(k+1)}{2}-\frac{t_c(t_c+1)}{2}\right] && \text{(by Equation 10.4.13 on page 424)} \\
&= \frac{r_c}{n_c}\binom{n_c}{r_c}\left[\binom{k+1}{2}-\binom{t_c+1}{2}\right] && \text{(by Equation 10.4.12 on page 424)} \\
&= Z\binom{n_c}{r_c}\left[\binom{k+1}{2}-\binom{t_c+1}{2}\right]. && \text{(by Equation 10.6.13 on page 456)}
\end{aligned}$$

At this point, all except for two of the quantities that we need to compute ASL@$k(V)$ are in place. The last two are the formulas for the sum of the ranks for a k-length sequence of non-relevant documents and the number of how many of them there are. The number of k-length sequences that only contain non-relevant documents is

$$\begin{aligned}\binom{n_c - (k - t_c)}{(n_c - r_c) - (k - t_c)} \left[\left(\sum_{i=1}^{c-1} r_i \right) = 0 \right] &= C[R_c = 0] \\ &= [R_c = 0]C.\end{aligned}$$

Finally, the sum of the ranks for these k-length sequences is

$$\begin{aligned}\binom{n_c - (k - t_c)}{(n_c - r_c) - (k - t_c)} (k+1) \left[\left(\sum_{i=1}^{c-1} r_i \right) = 0 \right] &= C(k+1)[R_c = 0] \\ &= [R_c = 0]C(k+1)\end{aligned}$$

because each sequence only contains a virtual document at rank $k + 1$.

Note that the values of both $[R_c = 0]C(k + 1)$ and $[R_c = 0]C$ are 0, if V_{pre} contains at least one relevant document, because the expression $[R_c = 0]$ evaluates to 1 when the number of relevant documents in V_{pre} is positive. If the number of documents in V_{pre} is 0, then $[R_c = 0]$ evaluates to 1.

Now, we have the information to compute ASL@$k(V)$. This value can be expressed as

$$\text{ASL@}k(V) = \frac{DY + EX + [R_c = 0]C(k+1)}{AY + BX + [R_c = 0]C}. \tag{10.6.14}$$

Again, it is important to emphasize that the expressions that appear after the $[R_c = 0]$ part, in both the numerator and denominator of this equation, are effectively ignored whenever any sequence in V_{pre} contains at least one relevant document.

10.6.3 Precision

The Definition of the Type-D Version of P@$k(V)$

The Type-D version of the precision measure at document cut-off k is typically defined so that its definition is equivalent to

$$\mathrm{P@}k(V) = \frac{\sum\limits_{i=1}^{k} I_R(v_i)}{k},$$

where V is a ranked vector of N documents for a query q and I_R is an indicator function for a collection R of relevant documents. Its value is equal to 1 if its argument represents a relevant document and is equal to 0, otherwise. This precision expression has the value 0 when there are no relevant documents among the first k documents.

The Derivation for the Type-T Version of P@$k(V)$

The Type-T version of the precision measure at document cut-off k is defined as

$$\mathrm{P@}k(V) = \left(R_c + \frac{(k - t_c)r_c}{n_c}\right)/k,$$

where the effective cut-off position and cut-off equivalence class index pair $(\tilde{k}, \tilde{c})$ is equal to (k, c_{k}).

10.6.4 Recall

The Derivation of the Type-D Version of R@$k(V)$

The Type-D version of the recall measure at document cut-off k is typically defined so that its definition is equivalent to

$$\mathrm{R}(V) = \left(\sum_{i=1}^{k} I_R(v_i)\right) / \left(\sum_{i=1}^{N} I_R(v_i)\right),$$

where k is not given explicitly, but is understood to be the number of retrieved documents. The variables k, q, N, R, V, and I_R have the same meanings as they did for the definitions of P@$k(V)$.

The Derivation for the Type-T Version of R@$k(V)$

The Type-T version of the recall measure at document cut-off k is defined as

$$\mathrm{R@}k(V) = \left(R_c + \frac{(k - t_c)r_c}{n_c}\right) / \left(\sum_{i=1}^{m} r_i\right),$$

where m is the number of equivalence classes in V for query q and the effective cut-off position and cut-off equivalence class index pair $(\tilde{k}, \tilde{c})$ is equal to (k, c_{k}). This recall expression is undefined when there are no relevant documents among the first m equivalence classes.

10.6.5 MZ-Based E Measure

The MZE measure assumes that there is at least one relevant document among $N = |V|$ documents in vector V. The definition below extends the typical definition so that the MZE measure is well-defined even when V does not contain any relevant documents.

$$\mathrm{MZE}(V) = \begin{cases} 1 - \dfrac{2}{P^{-1} + R^{-1}}, \text{ if } V \text{ has at least one relevant document} \\ \qquad\qquad\qquad\quad \text{among its first k documents;} \\ \qquad\qquad\qquad 1, \text{ otherwise.} \end{cases}$$

The variable P represents precision at point k and the variable R represents recall.

The Derivation for $\mathrm{MZE@}k(V)$

The Type-T version of the MZE measure at document cut-off k is defined as

$$\mathrm{MZE@}k(V) = \begin{cases} 1 - \frac{2}{(P@k(V))^{-1} + (R@k(V))^{-1}}, & \text{if } V \text{ has at least one relevant docu-} \\ & \text{ment among its first k documents;} \\ 1, & \text{otherwise.} \end{cases} \tag{10.6.15}$$

The variable $P@k(V)$ represents precision at point k and the variable $R@k(V)$ represents recall at this point. The effective cut-off position and cut-off equivalence class index pair $(\tilde{k}, \tilde{c})$ is equal to (k, c_{k}).

10.6.6 Reciprocal Rank

The reciprocal rank (RR) (Voorhees and Harman, 2005) is a measure that awards high values to ranking methods that rank relevant documents near the beginning of a ranking. Its values are in the range $[0, 1]$. The mean reciprocal rank (MRR) (Voorhees and Harman, 2005) measure is the variant that is the more well-known of the two. The MRR is the average of the reciprocal rank over multiple queries. Since the focus in this chapter is on comparing the performance of single queries, the reciprocal rank, rather than the mean reciprocal rank, is the metric that is used in the comparisons.

The Definition for the Type-D Version of RR@$k(V)$

The definition of the Type-D version of the reciprocal rank measure is the definition that is typically given in textbooks and the IR literature. The reciprocal rank at document cut-off value k on an ordered vector V of documents is defined as

$$\text{RR@}k(V) = \begin{cases} 1/i, \text{ if } \exists i \leq k, \text{ such that } V[i] \text{ is a relevant document, and} \\ \qquad \forall j < i, V[j] \text{ is a non-relevant document;} \\ 0, \text{ otherwise.} \end{cases}$$

This definition says that the value of the measure is the reciprocal of the rank of the relevant document in V that has the minimum rank among the first k documents. If such a document does not exist among the first k, then the value of the RR measure is 0.

Consider the following example in which there are three rows of ranked documents.

RRN
RNR
NRR

Assume that the document cut-off value is three (i.e., $k = 3$). The RR for each of the first two rows is $1^{-1} = 1/1 = 1$ because the first relevant document in each row is at rank 1. The RR for the third row is $2^{-1} = 1/2$ because its first relevant document occurs at rank 2.

Now, assume that $k = 1$. The RR values for the first two rows are unchanged at 1. However, the RR value for the third row is now 0 because all of its relevant documents occur at ranks higher than the cut-off point.

The Derivation for the Type-T Version of RR@$k(V)$

Figure 10.8 on the following page details several important relationships that are used to help derive an equation for this Type-T version of the RR@$k(V)$ measure. The passage below states how to compute the value for this version:

> To compute the tie-aware reciprocal rank, we first identify the first group V_i containing a relevant result. For each of the values j from $t_i + 1$ up to $\min(t_{i+1}, k)$, we compute the fraction of orderings in which the first relevant result occurs at exactly that position. Multiplying this fraction by $1/j$ and accumulating over j gives the correct answer. McSherry and Najork (2008)

Note that the variable i in the above quote identifies the index of the equivalence class E_i that contains the first relevant document. In the common notation that we have been using in this chapter, the role of the variable i in the quote is the same as the role of the variable c in our common notation. Therefore, document cut-off class E_i in the McSherry and Najork quote would be referred to as document cut-off class E_c in our notation. The effective cut-off position and cut-off equivalence class index pair $(\tilde{k}, \tilde{c})$ is determined as follows:

$$(\tilde{k}, \tilde{c}) = \begin{cases} (k, c_{\text{k}}), & \text{if } c_{\text{fr}} \geq c_{\text{k}}; \\ (t_{c_{\text{fr}}+1}, c_{\text{fr}}), & \text{otherwise.} \end{cases}$$

Before the derivation begins, it is helpful to provide an intuitive example to help conceptualize the process that occurs. This example has 11 documents and 3 equivalence classes. The first equivalence class has 2 documents, both are non-relevant; the second equivalence class has 5 documents of which only two are relevant; the third equivalence has just 4 documents, all of which are relevant. These combine for a total of

$$\binom{2}{2}\binom{5}{2}\binom{4}{4} = 1 \cdot 10 \cdot 1 = 10$$

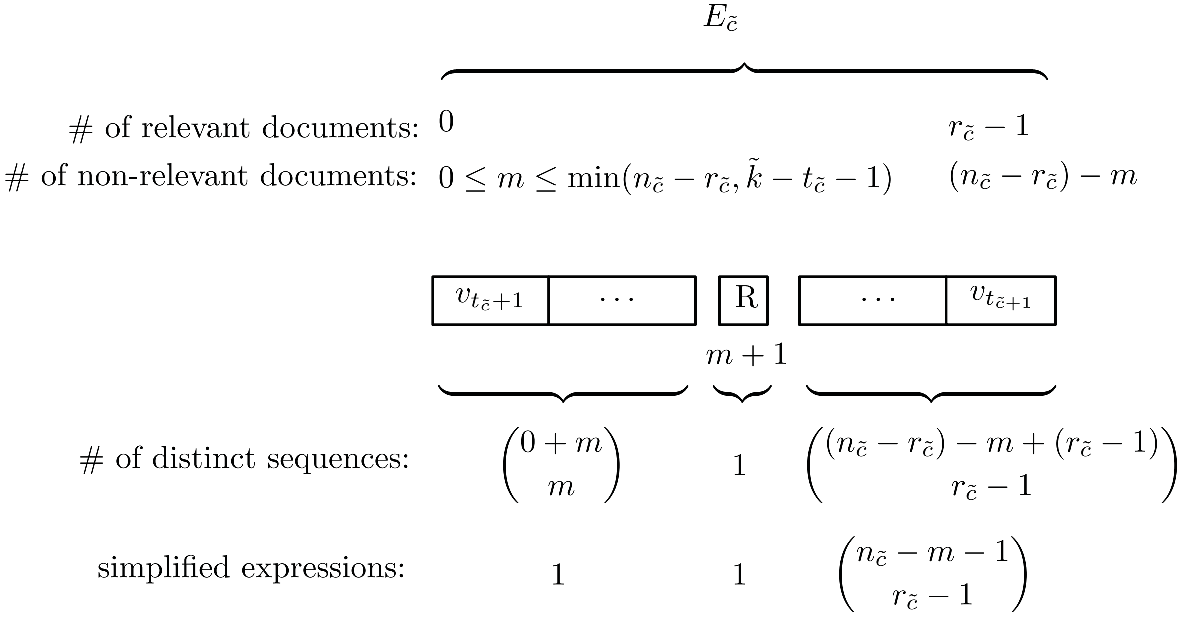

$E_{\tilde{c}}$
# of relevant documents: 0 $r_{\tilde{c}} - 1$
# of non-relevant documents: $0 \le m \le \min(n_{\tilde{c}} - r_{\tilde{c}}, \tilde{k} - t_{\tilde{c}} - 1)$ $(n_{\tilde{c}} - r_{\tilde{c}}) - m$
$v_{t_{\tilde{c}}+1}$ $\cdots$
R
$\cdots$ $v_{t_{\tilde{c}+1}}$
$m+1$
# of distinct sequences: $\binom{0+m}{m}$ 1 $\binom{(n_{\tilde{c}} - r_{\tilde{c}}) - m + (r_{\tilde{c}} - 1)}{r_{\tilde{c}} - 1}$
simplified expressions: 1 1 $\binom{n_{\tilde{c}} - m - 1}{r_{\tilde{c}} - 1}$

Figure 10.8: This diagram details the relationships that are associated with the cut-off class $E_{\tilde{c}}$ for the RR measure. Equivalence class $E_{\tilde{c}}$ contains a total of $n_{\tilde{c}}$ documents. It has $r_{\tilde{c}}$ relevant documents and $n_{\tilde{c}} - r_{\tilde{c}}$ non-relevant ones. The variable m denotes the number of consecutive non-relevant documents in subvector $V_{\tilde{c}}$ that precede the first relevant document in this subvector. These m non-relevant documents occupy positions $t_{\tilde{c}} + 1$ to $t_{\tilde{c}} + m$, inclusive, in $V_{\tilde{c}}$.

ways that these 11 documents can be ranked. These rankings can be viewed as rows in the ten-row-by-eleven-column table below.

NN	RRNNN	RRRR
NN	RNRNN	RRRR
NN	RNNRN	RRRR
NN	RNNNR	RRRR
NN	NRRNN	RRRR
NN	NRNRN	RRRR
NN	NRNNR	RRRR
NN	NNRRN	RRRR
NN	NNRNR	RRRR
NN	NNNRR	RRRR

Each row in the table represents one of the possible rankings and the columns represent the ranks. In order to make it easier to distinguish the document partitioning, according to their respective equivalence classes, the equivalence classes in each row are separated by several blanks. The first column in the table represents the documents, across all sequences, that are at rank 1; the second column represents the documents that are at rank 2, and so on, with the rightmost column representing those documents that are at rank 11.

In the previous example, the unstated assumption was that the documents in each of the three rankings had distinct RSVs. This means that each sequence of ranked documents was strongly ordered. That is, each of the documents was the only inhabitant of its equivalence class because there was exactly one sequence that was possible for each collection of N documents. In this next example, that is not the case because there were three distinct RSVs among the ranked documents. These resulted in three equivalence classes with cardinalities of 2, 5, and 4, respectively. Furthermore, these classes yielded

10 distinct sequences of ranked documents where, on the conceptual level, one possible ordering within an arbitrary equivalence class is considered to have the same importance as any other ordering within this class. In essence, it is best to view the documents within an equivalence class as being randomly ordered with each one having the same probability as any of the others to have a certain rank associated with it.

This example shows how to calculate the RR measure for these 10 rankings when the document cut-off window only has five slots (i.e., $k = 5$). Notice that the first relevant document occurs in the second equivalence class. Therefore, V_2 is the first subvector that contains a relevant document. Since $k = 5$ and the ranks in this subvector range from 3 to 7, inclusive, only its first $5 - 2 = 3$ ranks are of interest. From an inspection of the table, one can determine that 4 out of the 10 rows have a relevant document at rank 3. This corresponds to the situation where the number of non-relevant documents m in subvector V_2 that precede these four relevant ones is 0. This is important later when we derive the equation for RR@$k(V)$. The reciprocal rank for the first column in V_2 is 3^{-1} and the proportion of documents in that column that are relevant is 4/10. Together, these combine to give a value of partial RR of $4/10 \times 3^{-1}$ for that column. The partial RR values for the next two columns can be determined in a similar manner. The value for the second column in V_2 is $3/10 \times 4^{-1}$ because there are only three rows that have one non-relevant document (i.e., $m = 1$) at rank 3 that precedes a relevant document at rank 4. The value for the third column is $2/10 \times 5^{-1}$ because there are only two rows that have two non-relevant documents (i.e., $m = 2$) at ranks 3 and 4 that precede a relevant document at rank 5. With these values, the RR can be calculated as

$$\begin{aligned}\mathrm{RR@}k(V) &= \mathrm{RR@}5(V) \\ &= (4/10)3^{-1} + (3/10)4^{-1} + (2/10)5^{-1} \\ &= (4/10)(1/3) + (3/10)(1/4) + (2/10)(1/5)\end{aligned}$$

$$
\begin{aligned}
&= 4/30 + 3/40 + 2/50 \\
&= \frac{4 \cdot 20 + 3 \cdot 15 + 2 \cdot 12}{600} \\
&= \frac{149}{600} \\
&= 0.248333.
\end{aligned}
$$

Let c_{fr} denote the index of the equivalence class that the first relevant document is located in. Its definition is

$$
c_{\text{fr}} = \begin{cases} \min(S), & \text{if } \operatorname{card}(S) > 0; \\ \text{undefined}, & \text{otherwise}; \end{cases}
$$

where

$$
S = \left\{ i \,\middle|\, \left(\left(\sum_{1 \le p \le i} r_p \right) \ge 1 \right) \text{ and } 1 \le i \le m \right\}
$$

and m is the number of equivalence classes that are associated with vector V.

From the information in Figure 10.8 on page 465, and equating the value of c_{fr} with that for $\tilde{c}$, the value of the formula for $\text{RR@}k(V)$ can be written initially as

$$
\text{RR@}k(V) = \sum_{m=0}^{\min(n_{\tilde{c}} - r_{\tilde{c}}, \tilde{k} - t_{\tilde{c}} - 1)} \frac{\binom{0+m}{m} \binom{(n_{\tilde{c}} - r_{\tilde{c}}) - m + (r_{\tilde{c}} - 1)}{r_{\tilde{c}} - 1}}{\binom{n_{\tilde{c}}}{r_{\tilde{c}}}} \left(t_{\tilde{c}} + m + 1 \right)^{-1}. \qquad (10.6.16)
$$

We start the explanation of this equation by first discussing the summation range. The lower end of the range starts at 0 because the summing is over the number of consecutive non-relevant documents that can appear before the first relevant document (at rank $t_{\tilde{c}}+m+1$) in $V_{\tilde{c}}$. The upper end of the range is $\min(n_{\tilde{c}} - r_{\tilde{c}}, \tilde{k} - t_{\tilde{c}} - 1)$. The $n_{\tilde{c}} - r_{\tilde{c}}$ part is the number of non-relevant documents that are in $V_{\tilde{c}}$ and the $\tilde{k} - t_{\tilde{c}} - 1$ part is the maximum number of non-relevant documents that can be placed in the document cut-off window region of $V_{\tilde{c}}$. The "-1" in the last expression exists because there is a slot in this window that is reserved for the first relevant document in the ranking, this decreases the effective capacity of the window by 1 slot; hence, the "-1" adjustment was needed. The maximum number of non-relevant documents that can be placed in this window is the minimum of how many slots there are that are available for them (i.e., $\tilde{k} - t_{\tilde{c}} - 1$) and the number of non-relevant documents there are in $V_{\tilde{c}}$ (i.e., $n_{\tilde{c}} - r_{\tilde{c}}$).

The $\binom{0+m}{m}$ term represents the number of ways that the m non-relevant documents that precede the first relevant one can be arranged m at a time without regard to order. That can only occur one way because $\binom{0+m}{m} = \binom{m}{m} = 1$ when m is a natural number.

The $\binom{(n_{\tilde{c}}-r_{\tilde{c}})-m+(r_{\tilde{c}}-1)}{r_{\tilde{c}}-1}$ term represents the number of ways that the remaining $n_{\tilde{c}} - m - 1$ documents can be arranged after the m non-relevant ones that precede the first relevant ones have been chosen. These remaining documents consist of $r_{\tilde{c}} - 1$ relevant ones and $(n_{\tilde{c}} - r_{\tilde{c}}) - m$ non-relevant ones.

The $\binom{n_{\tilde{c}}}{r_{\tilde{c}}}$ term represents the number of distinct rankings in $V_{\tilde{c}}$ when it has $n_{\tilde{c}}$ total documents and $r_{\tilde{c}}$ of them are relevant. The $(t_{\tilde{c}}+m+1)^{-1}$ term represents the reciprocal rank at the column whose rank is $t_{\tilde{c}} + m + 1$.

The information above, plus other simplifications, allow Equation 10.6.16 on the preceding page to be rewritten as

$$\begin{aligned}\mathrm{RR@}k(V) &= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{m=0}^{\min(n_{\tilde{c}}-r_{\tilde{c}},\tilde{k}-t_{\tilde{c}}-1)} \binom{(n_{\tilde{c}} - r_{\tilde{c}}) - m + (r_{\tilde{c}} - 1)}{r_{\tilde{c}} - 1} (t_{\tilde{c}} + m + 1)^{-1} \\ &= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{m=0}^{\min(n_{\tilde{c}}-r_{\tilde{c}},\tilde{k}-t_{\tilde{c}}-1)} \binom{n_{\tilde{c}} - m - 1}{r_{\tilde{c}} - 1} (t_{\tilde{c}} + m + 1)^{-1}. \qquad (10.6.17)\end{aligned}$$

The McSherry and Najork Equations for RR@$k(V)$

McSherry and Najork (2008) provides a set of equations that, together, intend to calculate the RR@$k(V)$ value. The author of this dissertation found that these equations do not always calculate the correct value. From the description in their article, that appears below, of how the equations are supposed to work, this author found that the reason they do not work correctly is due to two typographical errors in the article.

The discussion of matters related to these equations proceed by providing two short examples of incorrect results, then showing how to correct the equations so that they provide the expected results. As further validation that the results are correct, the author shows how Equation 10.6.17 on the previous page and the corrected equations can cross-validate each other.

The quote below from the McSherry and Najork article lists the original equations, along with information that helps to rectify the typographical errors:

> We compute the fraction of orderings with the first relevant result at position $t_i + x$ by computing for each $t_i + x$ the fraction of orderings whose first x elements are *irrelevant,* and *then computing the difference between adjacent fractions.* Taking those orderings whose first x elements are relevant, minus those whose first $x + 1$ elements are irrelevant, gives the fraction whose first relevant element is at x. The fraction $f(x, r, n)$ of the orderings of r out of n relevant elements for which the first x are irrelevant follows as simple recursive definition:
>
> $$f(x, r, n) = \begin{cases} 1 - \dfrac{r}{n} & \text{if } x = 1 \\ \Bigg(1 - \dfrac{r}{n - x + 1})f(x - 1, r, n)\Bigg) & \text{otherwise[.]} \end{cases}$$
>
> Intuitively, each ordering that contributes to $f(x-1, r, n)$ will contribute to $f(x, r, n)$ if the next element is irrelevant, which occurs when none of the r relevant results are chosen from the set of $n - x + 1$ remaining results.
>
> Letting V_i be the first group containing a relevant result,
>
> $$\text{RR@}k(V) = \sum_{j=t_i+1}^{\min(t_{i+1}, \tilde{k})} \frac{f(j - t_i, r_i, n_i)}{j} [.] \qquad \text{(McSherry and Najork, 2008)}$$

The italicization in the first paragraph of the excerpt above did not appear in the article, it was added by the author of this dissertation to highlight the information as to how the article authors intended for their equations for RR@$k(V)$ to appear.

As was stated earlier, the equations, as given above in the article, do not compute the correct value for RR@$k(V)$. For instance, consider the scenario where there is a document collection with size $N = 3$ that has two relevant documents and one non-relevant one.

Also, assume that all of these documents have the same RSV. Hence, they all belong to the same equivalence class. The documents in this class can be arranged in only three orders. These orders are listed immediately below.

RRN
RNR
NRR

The correct RR@1(V), RR@2(V), and RR@3(V) values are, respectively, $2/3, 5/6$, and $5/6$. The values that are computed from the equation in the McSherry and Najork article are 1/3, 0, and 0, respectively.

The correction for this equation is straightforward. The key concept to notice is that the numerator of the RR@$k(V)$ equation should be changed from

$$f(j - t_i, r_i, n_i)$$

to

$$f(j - (t_i + 1), r_i, n_i) - f(j - t_i, r_i, n_i).$$

because of the “then computing the difference between adjacent fractions” passage in the McSherry and Najork article. Since this change means that the lower bound of the first parameter of the function f can now be 0, whereas, before, it was 1, a slight alteration needs to be made to function f. This alteration means changing the basis case of the recursion from 1 to 0. No other changes were necessary. The result of these changes are the revised equations below.

The Revised McSherry and Najork (2008) Equations for $\mathrm{RR@}k(V)$

$$f(x, r, n) = \begin{cases} 1, & \text{if } x = 0; \\ \left(1 - \dfrac{r}{n - x + 1}\right) f(x - 1, r, n), & \text{otherwise.} \end{cases} \tag{10.6.18}$$

$$\mathrm{RR@}k(V) = \sum_{j=t_i+1}^{\min(t_{i+1},k)} \frac{f(j - (t_i + 1), r_i, n_i) - f(j - t_i, r_i, n_i)}{j}. \tag{10.6.19}$$

The Revised McSherry and Najork Equations (Expressed in the Common Notation)

Equations 10.6.18 and Equation 10.6.19 use the variable i to denote the index of the first group that contains a relevant result. In essence, this is the index of the equivalence class of the first relevant document. In order to be consistent with previous notation, that used the variable c in the same role that the variable i is being used in in these equations, we define Equation 10.6.20 on the following page and Equation 10.6.21 on the next page to be the analogs of Equation 10.6.18 and Equation 10.6.19, respectively. The only differences between these two sets of equations are that the following substitutions were made to transform Equation 10.6.18 and Equation 10.6.19, respectively, into Equation 10.6.20 on the following page and Equation 10.6.21 on the next page:

$$\begin{aligned} t_{\tilde{c}} &\text{ for } t_i, \\ t_{c+1} &\text{ for } t_{i+1}, \\ n_{\tilde{c}} &\text{ for } n_i, \text{ and} \\ r_{\tilde{c}} &\text{ for } r_i. \end{aligned}$$

The slightly rewritten, but equivalent, equations appear immediately below.

$$f(x,r,n) = \begin{cases} 1, & \text{if } x = 0; \\ \left(1 - \dfrac{r}{n-x+1}\right) f(x-1,r,n), & \text{otherwise.} \end{cases} \tag{10.6.20}$$

$$\text{RR@}k(V) = \sum_{j=t_{\tilde{c}}+1}^{\min(t_{\tilde{c}+1},k)} \frac{f(j-(t_{\tilde{c}}+1), r_{\tilde{c}}, n_{\tilde{c}}) - f(j-t_{\tilde{c}}, r_{\tilde{c}}, n_{\tilde{c}})}{j} \tag{10.6.21}$$

Lemma 10.6.1. *Equation 10.6.20 and Equation 10.6.21, taken jointly, are equivalent to Equation 10.6.17 on page 469.*

Proof. Earlier, it was stated that $f(x,r,n)$ is the fraction of orderings that have x non-relevant documents in the first x slots of $V_{\tilde{c}}$. The first step in this proof is to develop a non-recursive expression for $f(x,r,n)$.

We notice that, for any positive natural number x, the function f recurses x times, with the values for its successive invocations starting at x and decreasing to 1. This means that successive values of the $n-x+1$ part of the Equation 10.6.20 fit the pattern $n-x+1, n-x+2, \cdots, n-1, n$. With these observations, we can write

$$f(x,r,n) = \prod_{i=1}^{x} \left(1 - \frac{r}{n-i+1}\right). \tag{10.6.22}$$

The second step in this proof is to argue that the recursive and non-recursive versions of function f always yield the same output value when they are presented with the same values for their input parameters. The manner in which the recursion was unrolled guarantees that the recursive and non-recursive versions of f are equivalent. Hence, Equation 10.6.22 yields the same values for its version of function f that Equation 10.6.20 yields for its version of function f when they are invoked with identical input values for their corresponding arguments. That is, both f functions yield the value 1 when x has the value 0, both yield the value 0 when $x > n-r$, and both also yield identical values when $1 < x \leq n - r$.

The third step in this proof is to find a combinatorial equivalent of Equation 10.6.22. We start by expressing $f(x, r, n)$ in terms of factorials.

$$\begin{aligned}
f(x,r,n) &= \prod_{i=1}^{x} \left(1 - \frac{r}{n-i+1}\right) \\
&= \prod_{i=1}^{x} \left(\frac{n-i+1-r}{n-i+1}\right) \\
&= \frac{(n-r)(n-r-1)\cdots(n-r-(x-1))}{n(n-1)\cdots(n-(x-1))} \\
&= \frac{\frac{(n-r)!}{(n-r-x)!}}{\frac{n!}{(n-x)!}} \\
&= \frac{(n-r)!}{(n-r-x)!}\frac{(n-x)!}{n!}. \qquad (10.6.23)
\end{aligned}$$

From an inspection of Equation 10.6.23, and experience with manipulating factorials, we notice that we can simplify matters by multiplying $f(x, r, n)$ by $\binom{n}{r}$. Once we do that, we obtain

$$\begin{aligned}
\binom{n}{r} f(x,r,n) &= \binom{n}{r}\frac{(n-r)!}{(n-r-x)!}\frac{(n-x)!}{n!} \\
&= \frac{n!}{r!(n-r)!}\frac{(n-r)!}{(n-r-x)!}\frac{(n-x)!}{n!}. \qquad (10.6.24)
\end{aligned}$$

From the inspection of this equation, we see that the $n!$ and $(n-r)!$ terms cancel out. This allows us to rewrite Equation 10.6.24 as

$$\begin{aligned}
\binom{n}{r} f(x,r,n) &= \frac{n!}{r!(n-r)!}\frac{(n-r)!}{(n-x-r)!}\frac{(n-x)!}{n!} \\
&= \frac{1}{r!}\frac{1}{(n-x-r)!}\frac{(n-x)!}{1} \\
&= \frac{(n-x)!}{r!(n-x-r)!}
\end{aligned}$$

$$= \binom{n-x}{r}.$$

The terms in this equation can be rearranged to yield

$$f(x,r,n) = \binom{n}{r}^{-1} \binom{n-x}{r}. \tag{10.6.25}$$

Before proceeding further, we restate Equation 10.6.17 on page 469 below for the convenience of the reader:

$$\mathrm{RR@}k(V) = \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{m=0}^{\min(n_{\tilde{c}}-r_{\tilde{c}}, \tilde{k}-t_{\tilde{c}}-1)} \binom{n_{\tilde{c}}-m-1}{r_{\tilde{c}}-1} \left(t_{\tilde{c}}+m+1\right)^{-1}.$$

This equation is our combinatorial version of the RR@$k(V)$ measure. We first presented it on page 469.

Our next goal in this proof is to use Equation 10.6.25 to help show that Equation 10.6.20 on page 473 and Equation 10.6.21 on page 473, taken jointly, are equivalent to Equation 10.6.17 on page 469. We continue in the following way.

$$\begin{aligned}
\mathrm{RR@}k(V) &= \sum_{j=t_{\tilde{c}}+1}^{\min(t_{\tilde{c}+1},k)} \frac{f(j-(t_{\tilde{c}}+1), r_{\tilde{c}}, n_{\tilde{c}}) - f(j-t_{\tilde{c}}, r_{\tilde{c}}, n_{\tilde{c}})}{j} \\
&= \sum_{j=t_{\tilde{c}}+1}^{\min(t_{\tilde{c}+1},k)} \left(f(j-(t_{\tilde{c}}+1), r_{\tilde{c}}, n_{\tilde{c}}) - f(j-t_{\tilde{c}}, r_{\tilde{c}}, n_{\tilde{c}})\right) j^{-1} \\
&= \sum_{j=0}^{\min(t_{\tilde{c}+1},k)-(t_{\tilde{c}}+1)} \left(f(j, r_{\tilde{c}}, n_{\tilde{c}}) - f(j+1, r_{\tilde{c}}, n_{\tilde{c}})\right) \left(j+t_{\tilde{c}}+1\right)^{-1} \\
&= \sum_{j=0}^{\min(t_{\tilde{c}+1},k)-(t_{\tilde{c}}+1)} \left(f(j, r_{\tilde{c}}, n_{\tilde{c}}) - f(j+1, r_{\tilde{c}}, n_{\tilde{c}})\right) \left(j+t_{\tilde{c}}+1\right)^{-1} \\
&= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{j=0}^{\min(t_{\tilde{c}+1},k)-(t_{\tilde{c}}+1)} \left(\binom{n_{\tilde{c}}-j}{r_{\tilde{c}}} - \binom{n_{\tilde{c}}-(j+1)}{r_{\tilde{c}}}\right) \left(j+t_{\tilde{c}}+1\right)^{-1}
\end{aligned}$$

$$
\begin{aligned}
&= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{j=0}^{\min(t_{\tilde{c}+1},k)-(t_{\tilde{c}}+1)} \left(\binom{n_{\tilde{c}}-j}{r_{\tilde{c}}} - \binom{n_{\tilde{c}}-(j+1)}{r_{\tilde{c}}} \right) (j+t_{\tilde{c}}+1)^{-1} \\
&= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{j=0}^{\min(t_{\tilde{c}+1},k)-(t_{\tilde{c}}+1)} \binom{n_{\tilde{c}}-j-1}{r_{\tilde{c}}-1} (j+t_{\tilde{c}}+1)^{-1}. \qquad (10.6.26)
\end{aligned}
$$

This proof is just about finished now. The main items to take care of are simplifying the summation limit and changing the summation index from j to m in Equation 10.6.26. Once these items have been taken care of, it is going to be evident that Equation 10.6.20 on page 473 and Equation 10.6.21 on page 473, taken jointly, are equivalent to Equation 10.6.17 on page 469.

We start this final effort by stating that

$$
\begin{aligned}
\min(t_{\tilde{c}+1},k)-(t_{\tilde{c}}+1) &= \min(t_{\tilde{c}+1}-(t_{\tilde{c}}+1), k-(t_{\tilde{c}}+1)) \\
&= \min(n_{\tilde{c}}-1, k-t_{\tilde{c}}-1)
\end{aligned}
$$

because $t_{\tilde{c}+1}$ is the index of the last element in V_c and $t_{\tilde{c}}$ is the index of the last element of $V_{\tilde{c}-1}$. This means that $n_{\tilde{c}} = t_{\tilde{c}+1} - t_{\tilde{c}}$. Therefore, we have

$$
\begin{aligned}
\mathrm{RR@}k(V) &= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{j=0}^{\min(t_{\tilde{c}+1},k)-(t_{\tilde{c}}+1)} \binom{n_{\tilde{c}}-j-1}{r_{\tilde{c}}-1} (j+t_{\tilde{c}}+1)^{-1} \\
&= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{j=0}^{\min(n_{\tilde{c}}-1,k-t_{\tilde{c}}-1)} \binom{n_{\tilde{c}}-j-1}{r_{\tilde{c}}-1} (j+t_{\tilde{c}}+1)^{-1} \\
&= \binom{n_{\tilde{c}}}{r_{\tilde{c}}}^{-1} \sum_{m=0}^{\min(n_{\tilde{c}}-r_{\tilde{c}},k-t_{\tilde{c}}-1)} \binom{n_{\tilde{c}}-m-1}{r_{\tilde{c}}-1} (m+t_{\tilde{c}}+1)^{-1} \qquad (10.6.27)
\end{aligned}
$$

because the summation's binomial term vanishes when $m > n_{\tilde{c}} - r_{\tilde{c}}$. By various transformations, we have just derived the same equation as Equation 10.6.17 on page 469. This completes the proof. □

10.7 Operationalizing What It Means For One Document Ranking to be Better Than Another Document Ranking

Based on the work in the previous sections of this chapter, we can now define, for the purposes of the research question that this chapter addresses, what it means to be able to state when one document ranking is better than another document ranking. Without loss of generality, assume that the performance values that are being plotted have been normalized so that they range from 0 to 1, inclusive, that is, the normalized values are in the closed interval $[0, 1]$. Therefore, the horizontal and vertical axes of the agreement and disagreement plots only need to range in value from 0 to 1, inclusive; the area that is covered by the plot is a unit square. Assume that the two measures are denoted by Measure M_1 and Measure M_2, and that the document rankings are denoted by Ranking R_1 and Ranking R_2.

Let A_1 denote the set of areas for document ranking R_1 where, such that for any $a_1 \in A_1$, one measure indicates that either performance is increasing, or is staying the same, for area a_1, and that the other measure indicates, for this same area a_1, that performance is increasing. We proceed similarly for document ranking R_2. Let A_2 denote the set of areas for document ranking R_2 where, such that for any $a_2 \in A_2$, one measure indicates that either performance is increasing, or is staying the same, for area a_2, and that the other measure indicates, for this same area a_2, that performance is increasing.

Each area in A_1 does not overlap with any other area in A_1, nor does any area in A_2 overlap with any other area in A_2. Let the sums of the areas in A_1 be denoted by s_1. Since the agreement-disagreement plot area constitutes a unit square, the value of s_1 is simply the proportion of the unit square that the areas in A_1 occupy. Similarly, let the sums of the areas in A_2 be denoted by s_2. The value of s_2 is the proportion of the unit square for ranking R_2 that the areas in A_2 occupy.

Now, we can state that we consider a document ranking R_1 to be better than a document ranking R_2 when the value of s_1 is greater than the value for s_2. In all other situations, the document ranking R_1 is not considered to be better than document ranking R_2.

10.8 Validation

Synthetic document collections of size 16 were constructed. Each collection had four equivalence classes with each class containing four documents. Each equivalence class could have zero to four relevant documents, inclusive, independent of the relevant document distributions that were associated with any of the other three classes. This resulted in 5 choices for each equivalence class (EC) because the class size was fixed at 4 and the number of relevant documents was allowed to vary from 0 to 4, inclusive, in each class. Overall, the number of possible EC combinations was $5^4 = 625$.

For each of these combinations, the document cut-off value was varied from 1 to 16 (i.e., the collection size), inclusive. This provided a total of $16 \times 625 = 10,000$ unique combinations of equivalence classes and document cut-off values to test for each performance measure.

Mathematica® (Wolfram, 2003) programs were developed to inspect each of the 625 EC combinations, by use of brute force techniques, and to calculate the values for the ASL@$k(V)$, MZE@$k(V)$, RR@$k(V)$, P@$k(V)$, and R@$k(V)$ performance measures. These values were computed for each of the 16 possible document cut-off points. These performance measure values, that were determined by brute force techniques, were later compared to their analytically-determined counterparts to verify that they all matched.

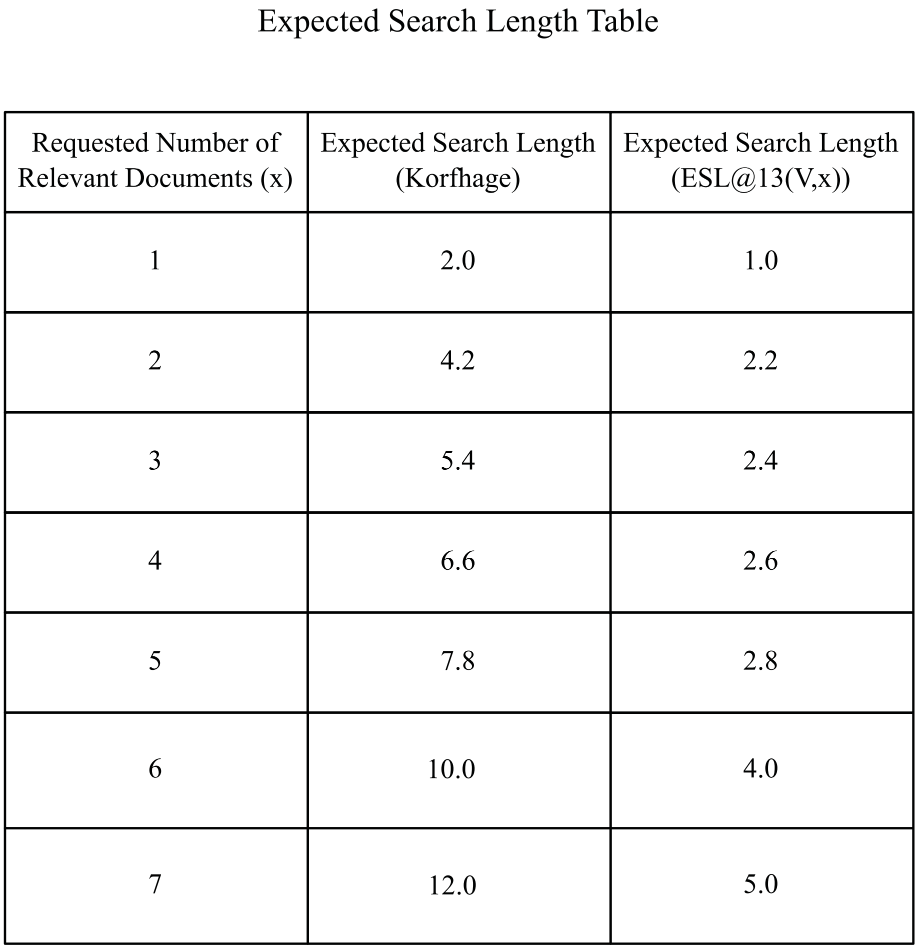
The validation process for the ESL@$k(V, x)$ measure was similar. In addition to everything that took place to verify the 5 measures that were just previously mentioned, a variable x was introduced to represent the requested number of relevant documents. The value of this variable was varied from 0 (no documents were requested) to 17 (this number is one more than the size of the document collection). The lower bound represented a request that could always be satisfied, the higher bound represented a request that could never be satisfied. The data generation process that was just described resulted in a total of $18 \times 10,000 = 18,000$ combinations of equivalence classes, document cut-off values, and requested numbers of relevant documents to examine. Brute force techniques were used to do the examination. The performance measure values that were obtained this way were compared to their analytically-determined counterparts to verify that they all matched.

In addition to the validation process for the ESL@$k(V, x)$ performance measure, that was described in the immediately previous paragraph, the author of this dissertation checked to see if the results that were calculated by this measure were the same as the results that appeared in a table labeled "Table 8.4 (Expected Search Length Table)" on page 206 in Korfhage (1997). For the convenience of the reader, this table appears below as part of Figure 10.9 on page 481. The information in this table was based on a set of documents that had three equivalence classes, namely, E_1, E_2, and E_3. The E_1 equivalence class has three documents (only 1 is relevant), E_2 has five documents (4 of them are relevant), and E_3 also has five documents (but only 2 are relevant). For our Type-T version of the ESL measure, $k = 13$ because the number of documents in the collection was 13, and the requested number of relevant documents ranged from 1 to 7, inclusive. The second column of the table contains the values that were calculated in Korfhage (1997). The third column of the table contains the corresponding values that were calculated by our Type-T version of the ESL. Note that, on first glance, the values appear to be different. The reason for this is that there are two common definitions of the ESL. This was discussed in Section 2.1.1 on page 15. The Cooper (1968) definition has as its value the mean number of *non-relevant* documents that must be examined in order to retrieve a specified number of relevant documents. But, the Korfhage (1997) definition defines the expected search length as the mean number of non-relevant *and relevant* documents that must be examined in order to retrieve a specified number of relevant documents. There is a simple mapping between a Korfhage ESL value and the ESL@$k(V, x)$ performance measure: either add the requested number of relevant documents to the ESL@$k(V, x)$ value or subtract the number from the Korfhage ESL value. For example, the Korfhage ESL value for three requested documents in Figure 10.9 on the next page is 5.4. The equivalent ESL@$k(V, x)$ value is $5.4 - 3 = 2.4$. If we apply either of these mappings to the expected search length values in the table of Figure 10.9 on the following page, the corresponding values can be shown to be equivalent.

10.9 Example: Comparing Type-T and Type-D Versions of the ASL Measure

Section 10.2.2 discussed how many performance measures implicitly assume that ranked vectors of documents are strongly-ordered. Figure 10.2 on page 411, and the discussion that it was a part of, demonstrated how variability (and overestimation and underestimation of the true value) of the ASL could occur. Figure 10.10 on page 482 illustrates the overestimation and underestimation that can occur when the document ranking is weakly-ordered, but the performance measure (e.g., ASL) implicitly assumes that the document ranking is strongly-ordered.

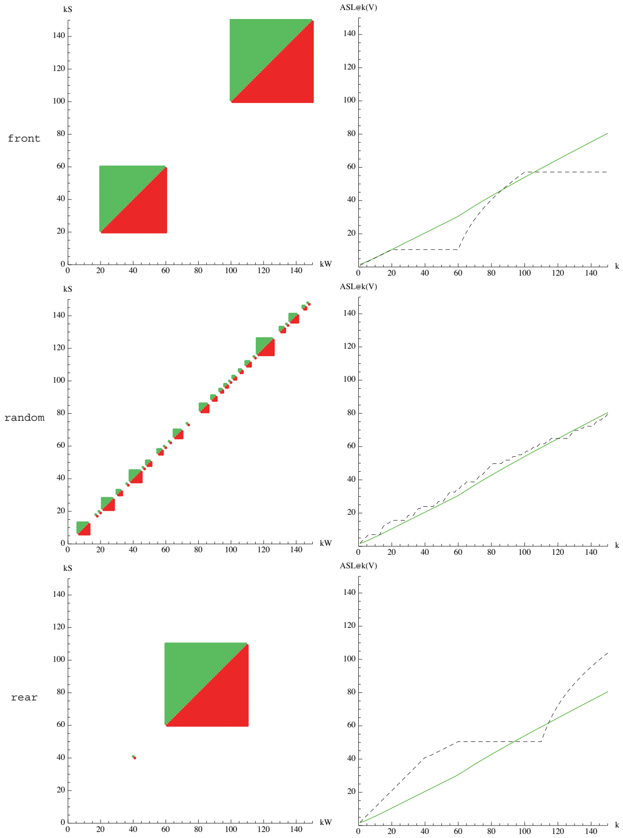
The document collection in the example for this section is of size 150 and the ranked vector V of documents has exactly two subvectors because there are only two distinct RSVs in the collection of documents. This means that there are two equivalence classes



Expected Search Length Table

Requested Number of Relevant Documents (x)	Expected Search Length (Korfhage)	Expected Search Length (ESL@13(V,x))
1	2.0	1.0
2	4.2	2.2
3	5.4	2.4
4	6.6	2.6
5	7.8	2.8
6	10.0	4.0
7	12.0	5.0

Figure 10.9: The information in this table is based on a set of documents that has three equivalence classes, namely, E_1, E_2, and E_3. The E_1 equivalence class has three documents (only 1 is relevant), E_2 has five documents (4 of them are relevant), and E_3 also has five documents (but only 2 are relevant). The values in the second column are based on a definition of the ESL that counts the mean number of relevant and non-relevant documents that must be retrieved in order to retrieve a requested number of relevant documents. The values in the third column are based on a definition of the ESL that only counts the mean number of relevant documents that must be examined before the requested number of relevant documents are retrieved. The values in the second and third columns are equivalent. For any designated row, the value in the second column can be converted to the value in the third column by subtracting the requested number of relevant documents, for this row, from it.



		ASL$_D$		
		worse	same	better
ASL$_T$	worse	9080	2095	0
	same	0	150	0
	better	0	2095	9080

		ASL$_D$		
		worse	same	better
ASL$_T$	worse	10971	204	0
	same	0	150	0
	better	0	204	10971

		ASL$_D$		
		worse	same	better
ASL$_T$	worse	9899	1276	0
	same	0	150	0
	better	0	1276	9899

Figure 10.10: The darkened areas in the plots of first column indicate the areas of disagreement, according to an extended version of the Losee (2000) comparison method, between a Type-T version of the ASL measure (i.e., ASL$_T$) and a Type-D version of the ASL(i.e., ASL$_D$) on the same collection of 150 documents. The green areas (located above the diagonal that starts at (0,0) and goes to (150,150)) in these left-column plots represent a region where the Type-T version of the ASL indicates that performance is decreasing but the Type-D version indicates that performance is staying the same. The red areas (located below the diagonal that starts at (0,0) and goes to (150,150)) in the left-column plots represent a region where the Type-T version of the ASL measure indicates that performance is increasing but the Type-D version indicates that performance is staying the same. The Type-D version always assumes that the vector V of ranked documents is strongly-ordered. The second column shows plots of the Type-T version of the ASL measure against Type-D versions. The solid green line represents the ASL values that were computed by the Type-T version of the ASL measure and the dashed black line represents the ASL values that were computed by the Type-D version of the ASL measure. The third column is a table which contains the distributions of the number of plot points that fall into each of the 9 categories that appear in Figure 10.1 on page 399(b). The first, second, and third rows of plots in this figure correspond, respectively, to the situations where the relevant documents in a vector V are at the front of each of its subvectors, are randomly-distributed within each of its subvectors, and are at the rear of each of its subvectors. Section 10.9 contains a detailed discussion of this figure.

and, hence, two subvectors. The subvector V_1 (associated with the higher-ranked equivalence class E_1) has sixty documents (20 of these are relevant) whereas the subvector V_2 (associated with the the lower-ranked class E_2) has ninety documents (40 of these are relevant).

The number of ways that the documents in V_1 can be sequenced is

$$\binom{60}{20}$$

and the number of sequences for V_2 is

$$\binom{90}{40}.$$

Since the documents in each of these two subvectors can be arranged independently of those in the other subvector, the joint number j of possible sequences is

$$\binom{60}{20}\binom{90}{40},$$

which is approximately 2.51×10^{41}. If we assume that each sequence is equally likely, then a correctly-implemented algorithm for the Type-T version of the ASL computes the same mean value m for vector V, no matter which of these j sequences is used as an input to the algorithm. Conceptually, the way that the algorithm does this is to, first, determine the ASL value for each sequence and, then, calculate the mean of these individual sequence-specific ASL values. The resultant value is m.

In other words, a Type-T version of the ASL is not concerned with the calculation of the ASL value for any one particular sequence of documents because all the sequences that are associated with a given equivalence class are the same, from the perspective of how many relevant documents they contain, and the calculated value for a specific sequence may not be representative of the ASL value for the equivalence classes as a whole. Instead, the value that an algorithm for a Type-T version of the ASL calculates is based on the joint number j of all the $|V|$-length sequences that are possible for the documents in vector V. This makes its value independent of the sequence that is associated with a particular ranking and implies that the computed ASL value is stable over the possible sequences and is neither an overestimation nor an underestimation of the ASL value for the ranked vector V of documents. Note that even though there are j equivalent V-length sequences for the collection of documents that is represented by V, the documents in this collection can only have one physical order at a time. A Type-D version of a performance measure algorithm essentially uses the given physical order. A Type-T measure views this physical order as just one of the j physical orders that are possible and bases its calculations not just on the given physical order but, also, on the other $j-1$ physical orders. It accomplishes this by determining the value of the measure for each other these j orders and, then, reporting the mean of these values as its result.

Analysis of the Plots

In the plots of Figure 10.10 on page 482, the gray areas (located above the diagonal that starts at (0,0) and goes to (150,150)) in these left-column plots represent a region where the Type-T version of the ASL indicates that performance is decreasing but the Type-D version indicates that performance is staying the same. The red areas (located below the diagonal that starts at (0,0) and goes to (150,150)) in the left-column plots represent a region where the Type-T version of the ASL measure indicates that performance is increasing but the Type-D version indicates that performance is staying the same.

The solid green line in each of the middle-column plots represents the performance values that are calculated by the Type-T version (labeled ASL_t and on the horizontal axis) of the ASL measure. The black dashed lines represent the values that are calculated by the Type-D version (labeled ASL_{nt} and on the vertical axis) of this measure.

The matrices in the right-column of each row of Figure 10.10 on page 482 contain the distributions of the number of plot points that fall into each of the 9 joint categories that appear in Figure 10.1(b) on page 399. The rows in these matrices represent the values for Type-T versions of the ASL measure whereas the columns represent the values for Type-D versions of the ASL. The value at the intersection of a row and column represents the joint value for the Type-T ASL category and the Type-D column category. There are three categories for each dimension of a matrix: *worse* (the performance decreased for the measure between two given points a and b), *same* (the performance stayed the same between points a and b), and *better* (the performance increased between points a and b). In this figure, the left-hand side categories for a matrix represents those for the Type-T version of the ASL (i.e., ASL_{t}) measure and the categories that are listed across the top of the matrix are for the Type-D version of the ASL (i.e., ASL_{nt}).

The row that is labeled *front* in Figure 10.10 on page 482 is an example of how the ASL value can be underestimated. The r_1 relevant documents in subvector V_1 are positioned at the front of subvector V_1 (i.e., they occupy the first r_1 positions in V_1) and the r_2 relevant documents in subvector V_2 are positioned at the front of subvector V_2 (i.e., they occupy the first r_2 positions in V_2). This minimizes the ASL (i.e., performance is increased). Most of the document cut-off points for the middle-column plot of this row have horizontal coordinates where the vertical coordinates for the Type-D version of the ASL are less that those of the corresponding Type-T version of the ASL. The mean of the ASL values that are associated with the dashed line (e.g., calculated by the Type-D version of the ASL) is lower than the mean of the ASL values that are associated with the solid line (e.g., calculated by the Type-T version of the ASL). This indicates that the ASL is underestimated.

The row that is labeled *random* in Figure 10.10 on page 482 is an example where the documents are randomly-ordered within each subvector. Generally, in this case, the ASL is neither maximized nor minimized. Rather, its expected value should be the same value that would be calculated by a Type-T version of the ASL. The middle-column plot shows that there is much agreement between the performance values that are calculated by the Type-T and Type-D versions of the ASL measure for the ranked documents. The mean of the ASL values that are associated with the dashed line (e.g., calculated by the Type-D version of the ASL) is almost the same as the mean of the ASL values that are associated with the solid line (e.g., calculated by the Type-T version of the ASL). This indicates that the mean of the ASL values calculated by the Type-T version of the ASL measures is approximately equal to the mean ASL that is calculated by the Type-D measure when the documents within each subvector V_i are randomly ordered..

The row that is labeled *rear* in Figure 10.10 on page 482 is an example of how the ASL value can be overestimated. The r_1 relevant documents in subvector V_1 are positioned at the rear of subvector V_1 (i.e., they occupy the last r_1 positions in V_1) and the r_2 relevant documents in subvector V_2 are positioned at the rear of subvector V_2 (i.e., they occupy the last r_2 positions in V_2). This maximizes the ASL (i.e., performance is decreased). Most of the document cut-off points in its middle-column plot have horizontal coordinates where the vertical coordinates for the Type-D version of the ASL are greater that those of the corresponding Type-T version of the ASL. The mean of the ASL values that are associated with the dashed line (e.g., calculated by the Type-D version of the ASL) is higher than the mean of the ASL values that are associated with the solid line (e.g., calculated by the Type-T version of the ASL). This indicates that the ASL is overestimated.

It is important to note here that the Type-T version of the ASL should compute identical values for each of the vectors that are associated with the three rows in the figure because the number of equivalence classes are the same, the number of documents (and proportion of relevant documents) in their E_1 equivalence classes are the same, and the number of documents (and proportion of relevant documents) in their E_2 equivalence classes are the same.

However, for a Type-D version of the ASL, the sequences all appear to be different, even though they are not, because the rankings are weakly-ordered. Effectively, there are V equivalence classes for a Type-D version of the ASL measure. Each of these equivalence classes is of size 1. The only time that the performance value that would be calculated by a Type-D version of the ASL would approximate that for a Type-T version in this situation is when the documents are randomly ranked within their respective subvectors. This is illustrated by the second row of subfigures in Figure 10.10 on page 482.

10.10 Example: Comparing the ASL Measure With the MZE, ESL, and RR Measures

Figure 10.11 on the following page uses a small synthetic test collection of 50 documents (with certain characteristics that are described below) to illustrate that an extended version of the Losee (2000) method, in conjunction with Type-T versions of the ESL, MZE, and RR measures, can be used to obtain a better understanding of how the ASL performance measure compares with these measures for the best-case, coordination level matching, decision-theoretic, inverse document frequency, random case, and worse case ranking methods.

The gray areas in the plots represent regions where the ASL measure indicates that performance is decreasing but the measure that the ASL is being compared to indicates that performance is staying the same. The red areas represent regions where the ASL measure indicates that performance is increasing but the other measure indicates that performance is staying the same. The green areas represent regions where the ASL measure indicates that performance is decreasing but the other measure indicates

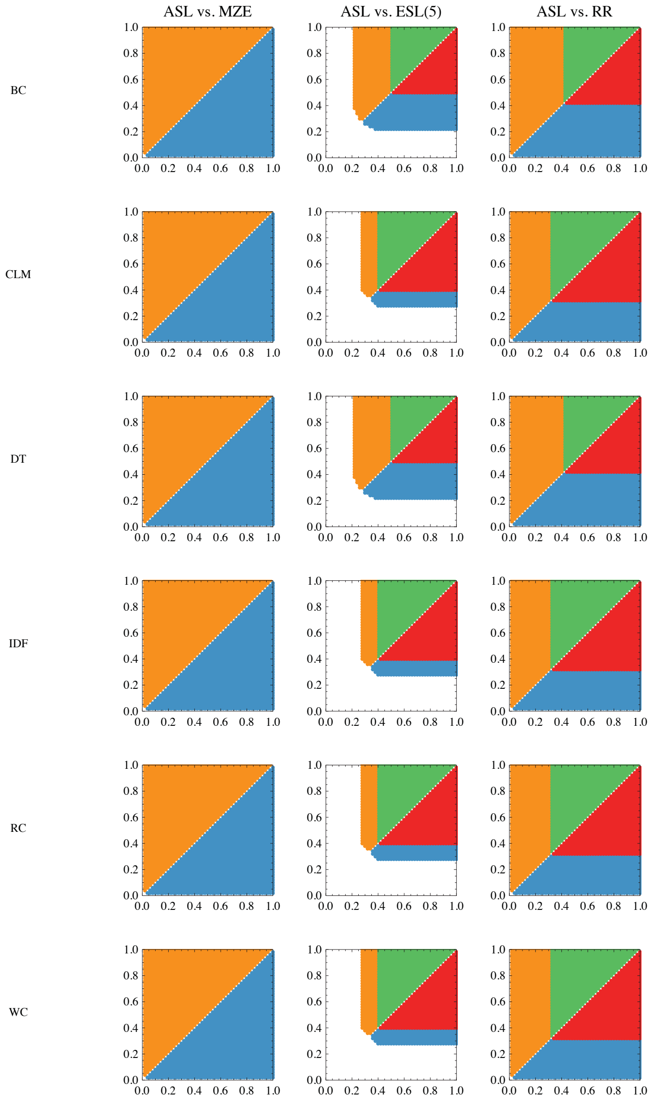


Figure 10.11: Areas of agreement and disagreement for the ASL measure when it is compared to the MZE, ESL, and RR performance measures for the BC, CLM, DT, IDF, RC, and WC ranking methods. The query-document collection has the characteristic $(r_1, r_0, s_1, s_0) = (5, 10, 15, 20)$. The horizontal and vertical axes represent the proportion of documents that have been examined at a certain point k in a ranking. The number of requested relevant documents used for the ESL measure is 5. The white areas represent regions of agreement, the darker areas represent regions of disagreement. The orange regions represent areas where the ASL measure indicates that performance is decreasing whereas the non-ASL measure indicates that performance is increasing. The blue regions represent areas where the ASL measure indicates that performance is increasing whereas the non-ASL measure indicates that performance is decreasing. The red regions represent areas where the ASL measure indicates that performance is increasing whereas the non-ASL measure indicates that performance did not change. The green regions represent areas where the ASL measure indicates that performance is decreasing whereas the non-ASL measure indicates that performance did not change.

that performance is increasing. Finally, the blue areas represent regions where the ASL measure indicates that performance is increasing but the other measure indicates that performance is decreasing.

The query-document collection description, in terms of the notation that was introduced in Chapter 2, Section 2.2.6, page 24, is $(r_1, r_0, s_1, s_0) = (5, 10, 15, 20)$. This states that 5 of the 15 relevant documents contain the query term whereas the other 10 relevant documents do not contain the query term. This description also indicates that the collection has 35 non-relevant documents, with 15 of them containing the query term and the other 20 not containing the query term.

The investigation of the rankings behind the plots in Figure 10.11 on the previous page showed that the 6 different ranking methods yielded only two equivalence classes for each ranking. Overall, there were two sets of equivalence classes, namely, sets S_{bd} (for the BC and DT ranking methods) and S_{cirw} (for the CLM, IDF, RC, WC ranking methods). Each of these sets has two members E_1 and E_2. For set S_{bd}, the E_1 equivalence class contained 30 documents (ten of them were relevant) and the E_2 equivalence class contained 20 documents (five of them were relevant). For set S_{cirw}, the E_1 equivalence class contained 20 documents (five of them were relevant) and the E_2 equivalence class contained 30 documents (ten of them were relevant).

The reason that there were so few distinct equivalence classes is due to constraints that were induced on the RSVs by the query-document model (i.e., binary relevance, binary feature frequency). Our model allowed for very little variation in the RSVs within an arbitrary ranking. The RSVs either had a value of 0 or some non-zero value $\bar{z}$. If the non-zero value was $\bar{z}$ for any document in the ranking, then all other documents in this ranking that had non-zero RSVs had the same value $\bar{z}$ for their RSV. The result was that the ranking methods, for non-empty collections, produced document orderings that had, at most, two equivalence classes.

The data that was used to produce the plots for the best-case and decision-theoretic ranking methods appear in Table 10.5 on the following page and the data that was used to produce the plots for the coordination-level matching, inverse document frequency, random case, and worst-case ranking methods appear in Table 10.6 on page 492.

Analysis of the Plots

The analyses for the BC and DT ranking methods were identical because their equivalence classes were both members of the set S_{bd}. Likewise, the analyses for the CLM, IDF, RC, and WC ranking methods were identical because their corresponding equivalence classes, though different from those of the ASL@$k(V)$ and RR@$k(V)$ measures, were also identical, because they were members of the set S_{cirw}. For the convenience of the reader, we restate the following: E_1, the higher-ranked equivalence class for the BC and DT ranking methods has 30 documents (ten of them are relevant) and E_2, the lower-ranked equivalence class, has 20 documents (five of them are relevant); the equivalence class E_1 for the CLM, IDF, RC, and WC ranking methods has 20 documents (five of them are relevant) and E_2 has 30 documents (ten of them are relevant).

We start our analyses by noticing that the plots indicated that the ASL@$k(V)$ and MZE@$k(V)$ measures, over all the ranking methods, disagreed on relative rankings much more than they agreed. The matrix at the end of the first row in Figure 10.12 on page 494 provided detailed distribution information on the agreement and disagreement values. There was one kind of agreement and two kinds of disagreement that were present in the plots. The amount of agreement was 2% and the amount of disagreement was 98%.

The plots indicated that the ASL@$k(V)$ measure and the ESL@$k(V, x)$ measure (when the requested number of relevant documents is 5), over all the ranking methods, disagreed on the relative rankings more than they agreed. In this example, for the ESL measure, we are interested in the mean number of non-relevant documents that must be retrieved

Table 10.5: Values of Selected Performance Measures For All Cut-off Points For Two Equivalence Classes. The higher ranked equivalence class has 30 documents, the lower-ranked one has 20 documents. The number of relevant documents in these classes are, respectively, 10 and 5.

k	ASL	MZE	ESL(5)	RR	k	ASL	MZE	ESL(5)	RR
1	1.66667	0.958333	1.	0.333333	26	13.5	0.577236	9.09091	0.555247
2	2.09375	0.921569	2.	0.448276	27	14.	0.571429	9.09091	0.555247
3	2.43846	0.888889	3.	0.500274	28	14.5	0.565891	9.09091	0.555247
4	2.79268	0.859649	4.	0.526273	29	15.	0.560606	9.09091	0.555247
5	3.18383	0.833333	4.99116	0.539872	30	15.5	0.555556	9.09091	0.555247
6	3.61062	0.809524	5.95402	0.547125	31	15.878	0.554348	9.09091	0.555247
7	4.06423	0.787879	6.86118	0.551011	32	16.2619	0.553191	9.09091	0.555247
8	4.53603	0.768116	7.68216	0.55308	33	16.6512	0.552083	9.09091	0.555247
9	5.01949	0.75	8.38879	0.554167	34	17.0455	0.55102	9.09091	0.555247
10	5.51013	0.733333	8.95997	0.554726	35	17.4444	0.55	9.09091	0.555247
11	6.00503	0.717949	9.38485	0.555006	36	17.8478	0.54902	9.09091	0.555247
12	6.50237	0.703704	9.6643	0.55514	37	18.2553	0.548077	9.09091	0.555247
13	7.00105	0.690476	9.81042	0.555203	38	18.6667	0.54717	9.09091	0.555247
14	7.50043	0.678161	9.8446	0.55523	39	19.0816	0.546296	9.09091	0.555247
15	8.00016	0.666667	9.79433	0.555241	40	19.5	0.545455	9.09091	0.555247
16	8.50005	0.655914	9.68948	0.555245	41	19.9216	0.544643	9.09091	0.555247
17	9.00002	0.645833	9.55865	0.555247	42	20.3462	0.54386	9.09091	0.555247
18	9.5	0.636364	9.42597	0.555247	43	20.7736	0.543103	9.09091	0.555247
19	10.	0.627451	9.30897	0.555247	44	21.2037	0.542373	9.09091	0.555247
20	10.5	0.619048	9.21764	0.555247	45	21.6364	0.541667	9.09091	0.555247
21	11.	0.611111	9.15476	0.555247	46	22.0714	0.540984	9.09091	0.555247
22	11.5	0.603604	9.11737	0.555247	47	22.5088	0.540323	9.09091	0.555247
23	12.	0.596491	9.09904	0.555247	48	22.9483	0.539683	9.09091	0.555247
23	12.	0.596491	9.09904	0.555247	48	22.9483	0.539683	9.09091	0.555247
25	13.	0.583333	9.09091	0.555247	50	23.8333	0.538462	9.09091	0.555247

Table 10.6: Values of Selected Performance Measures For All Cut-off Points For Two Equivalence Classes. The higher ranked equivalence class has 20 documents, the lower-ranked one has 30 documents. The number of relevant documents in these classes are, respectively, 5 and 10.

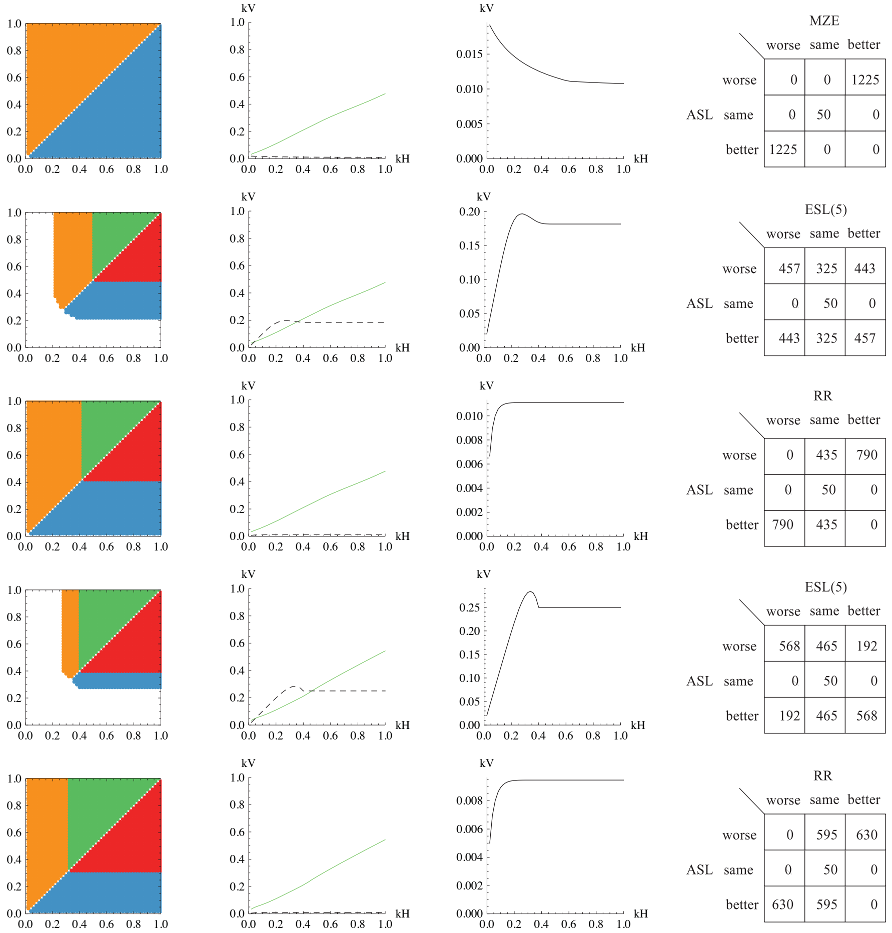
k	ASL	MZE	ESL(5)	RR	k	ASL	MZE	ESL(5)	RR
1	1.75	0.96875	1.	0.25	26	14.2143	0.658537	12.5	0.473252
2	2.2875	0.941176	2.	0.348684	27	14.7955	0.650794	12.5	0.473252
3	2.69466	0.916667	3.	0.399854	28	15.3696	0.643411	12.5	0.473252
4	3.04952	0.894737	4.	0.429201	29	15.9375	0.636364	12.5	0.473252
5	3.40249	0.875	4.99968	0.446809	30	16.5	0.62963	12.5	0.473252
6	3.77742	0.857143	5.998	0.45757	31	17.0577	0.623188	12.5	0.473252
7	4.18115	0.840909	6.99278	0.464158	32	17.6111	0.617021	12.5	0.473252
8	4.61208	0.826087	7.98013	0.468149	33	18.1607	0.611111	12.5	0.473252
9	5.06535	0.8125	8.95395	0.470514	34	18.7069	0.605442	12.5	0.473252
10	5.53553	0.8	9.90519	0.471869	35	19.25	0.6	12.5	0.473252
11	6.01768	0.788462	10.8212	0.472607	36	19.7903	0.594771	12.5	0.473252
12	6.50782	0.777778	11.685	0.472984	37	20.3281	0.589744	12.5	0.473252
13	7.00292	0.767857	12.4743	0.473157	38	20.8636	0.584906	12.5	0.473252
14	7.50083	0.758621	13.1607	0.473226	39	21.3971	0.580247	12.5	0.473252
15	8.00014	0.75	13.7087	0.473248	40	21.9286	0.575758	12.5	0.473252
16	8.5	0.741935	14.0748	0.473252	41	22.4583	0.571429	12.5	0.473252
17	9.	0.734375	14.2061	0.473252	42	22.9865	0.567251	12.5	0.473252
18	9.5	0.727273	14.0395	0.473252	43	23.5132	0.563218	12.5	0.473252
19	10.	0.720588	13.5	0.473252	44	24.0385	0.559322	12.5	0.473252
20	10.5	0.714286	12.5	0.473252	45	24.5625	0.555556	12.5	0.473252
21	11.1563	0.703704	12.5	0.473252	46	25.0854	0.551913	12.5	0.473252
22	11.7941	0.693694	12.5	0.473252	47	25.6071	0.548387	12.5	0.473252
23	12.4167	0.684211	12.5	0.473252	48	26.1279	0.544974	12.5	0.473252
24	13.0263	0.675214	12.5	0.473252	49	26.6477	0.541667	12.5	0.473252
25	13.625	0.666667	12.5	0.473252	50	27.1667	0.538462	12.5	0.473252

in order to retrieve 5 relevant documents. Our concise notation for that is ESL(5) and appears in Figure 10.11 on page 488, Table 10.5 on page 491, and Table 10.6 on the preceding page. The matrix at the end of the second row in Figure 10.12 on the next page provides detailed distribution information on the agreement and disagreement values for the BC and DT ranking methods. There were 3 kinds of agreement and 4 kinds of disagreement. The amount of agreement was 49.56% and the amount of disagreement was 50.44%. The matrix at the end of the fourth row in Figure 10.12 on the following page provided detailed distribution information on the agreement and disagreement values for the CLM, IDF, RC, and WC ranking methods. The amount of agreement was 47.44% and the amount of disagreement was 52.56%.

The plots for the ASL measures versus the RR measures, over all of the ranking methods, show that there was 1 kind of agreement and 4 kinds of disagreement. The matrices at the ends of the third and fifth rows in Figure 10.12 on the next page show that the amount of agreement was 2% and that the amount of disagreement was 98%. The only agreements occurred when both measures indicated that the performance was the same (i.e., did not change) between 2 points. The types of disagreements were the same for both sets of measures, the only difference was that the proportions of disagreement kinds had different values in the BC and DT set of ranking methods than they did in the CLM, IDF, RC, and WC set.

10.11 Summary

This chapter began with the development of a table of important characteristics to consider when comparing points of agreement and disagreement between ranking measures on the relevant ordering of documents. An example that involved the ASL@$k(V)$ measure illustrated how a Type-D version of a performance measure could provide different results than a Type-T version when there were duplicate RSVs in the collection of ranked



MZE

ASL		worse	same	better
	worse	0	0	1225
ASL	same	0	50	0
	better	1225	0	0

ESL(5)

ASL		worse	same	better
	worse	457	325	443
ASL	same	0	50	0
	better	443	325	457

RR

ASL		worse	same	better
	worse	0	435	790
ASL	same	0	50	0
	better	790	435	0

ESL(5)

ASL		worse	same	better
	worse	568	465	192
ASL	same	0	50	0
	better	192	465	568

RR

ASL		worse	same	better
	worse	0	595	630
ASL	same	0	50	0
	better	630	595	0

Figure 10.12: The leftmost column contains the 5 distinct plots from Figure 10.11 on page 488, the other columns contain more detailed information about each plot. The horizontal and vertical axes of the plots represent the proportion of documents that have been examined at a certain point k in a ranking. The horizontal axis always represents the proportional k value for the ASL@$k(V)$ measure and the vertical axis represents the proportional k value for a non-ASL measure. The graphs in the second column plot Type-T ASL values against non-ASL Type-T values. The third column provides more detail about the values from the non-ASL measure. This is important because some of the values from the non-ASL measure may differ by more than an order of magnitude from their ASL value counterparts (the graphs in the second column of the third and fifth rows are an example of this situation). The matrices in the fourth column provide distribution information about the values that were used to construct the plots in the first column. The orange regions represent areas where the ASL measure indicates that performance is decreasing whereas the non-ASL measure indicates that performance is increasing. The blue regions represent areas where the ASL measure indicates that performance is increasing whereas the non-ASL measure indicates that performance is decreasing. The red regions represent areas where the ASL measure indicates that performance is increasing whereas the non-ASL measure indicates that performance did not change. The green regions represent areas where the ASL measure indicates that performance is decreasing whereas the non-ASL measure indicates that performance did not change.

documents.

Later, we discussed how to reasonably extend the performance measures so that each measure was defined at every point in a ranking. We also discussed weak and strong orders and what "rank" means when the order of the ranked documents is weak rather than strong. This chapter extended the general framework for handling ties that appeared in McSherry and Najork (2008) and introduced the notions of an effective document cut-off equivalence class and an effective document cut-off point.

Combinatoric-based versions of the ASL@$k(V)$, MZE@$k(V)$, RR@$k(V)$, P@$k(V)$, R@$k(V)$, and ESL@$k(V, x)$ measures were developed. An error in the equation for the ESL@$k(V, x)$ performance measure, due to a typographical error that occurred in the McSherry and Najork (2008) article, was pointed out along with a suggested correction. We discussed how the analytic versions of the ASL, ESL, MZE, RR, recall, and precision measures were validated.

This chapter concluded with two examples that used plots and distribution matrices to illustrate how well the ASL measure agreed with the ESL, MZE, and RR measures on relative rankings *for the data that was used in the examples.* The first plot compared results from the Type-T version of the ASL performance measure with results from three distributions of the same data that assumed a strong ordering of the ranked documents. The second example involved plots that helped illustrate comparisons of the ASL measure with the MZE, ESL, and RR performance measures.

How well one performance measure compares with another over a vector of ranked documents is a function of the characteristics of the two measures, the ranking methods, the granularity of the RSVs, the number of equivalence classes, and the numbers of relevant and non-relevant documents within these classes. The complexity of the measure comparison problem makes it very difficult to issue a general statement about how one measure compares with another one. Near the end of this chapter, we operationalized what it means, in the context of this chapter, for one document ranking to be considered better than another document ranking.

The Type-T measures that were developed in this chapter, in conjunction with the extended version of the Losee comparison method, make important contributions to IR performance evaluation in that they can be used by researchers to help study, obtain more insight, and provide a better understanding of the interactions between the factors that influence how one measure compares with another one for both specific rankings and collections of rankings.

Chapter 11

Summary and Conclusions

This chapter summarizes the work that was discussed in the prior chapters, details how the research was conducted, discusses difficulties that were encountered, and states the significant findings. Near the end of this chapter, we discuss some implications of this research, make some recommendations for future researchers who may want to follow paths similar to those that were taken for this research, and discuss several possibilities for extending this research. The most significant aspect of this research was that the author was able to formulate a theory with respect to information retrieval performance measures and document ranking methods, centered mainly around composition theory, partition theory, and enumerative combinatorics; develop a model for this theory; and empirically validate that the model produced the expected theoretical results.

11.1 Goals

The primary goal of this research was to investigate the use of combinatorics in the development of equations for the Average Search Length (ASL) (Equation 2.0.1 on page 12) performance measure and its independent variables, namely, the normalized average position of a relevant document ($\mathcal{A}$) (Equation 2.0.2 on page 13) and the quality of a ranking method ($\mathcal{Q}$) in a centralized information retrieval context. This research also compared the performance, as measured by the ASL, with the performances as measured by the

MZ-based E measure (MZE) (Equation 2.1.4 on page 17) (van Rijsbergen, 1979), the Expected Search Length (ESL) (Equation 2.1.1 on page 15) (Cooper, 1968), and the Mean Reciprocal Rank (MRR) (Equation 2.1.3 on page 17) (Voorhees, 2001) measures. Due to the fact that the MRR was being calculated for a single query, as contrasted with a set of queries, the reciprocal rank (RR) (Equation 2.1.2 on page 16) measure was used instead of the MRR because these two measures provide identical results when the set of queries for the MRR only contains one query.

A secondary goal of this research was to demonstrate that the resultant equations that were developed for this analytic approach produced results that were statistically the same as the corresponding results that were obtained by empirical means. The author of this dissertation was successful in attaining all of these goals and in finding answers for each the three research questions that are enumerated below.

11.2 Questions

1. What would be the characteristics of a combinatoric measure (CM_ASL), based on the ASL, that performs the same as a probabilistic measure of retrieval performance, also based on the ASL?

2. Does the CM_ASL measure produce the same performance results as that of an actual document ranking? [In other words, is there any statistically significant difference between the predicted performance and the performances observed in actual rankings?]

3. When does the ASL measure and one of these measures (i.e., MZE, ESL, and MRR) both imply that one document ranking is better than another document ranking?

11.3 Steps

The steps that were taken to obtain answers for each of these questions began with the author recognizing that weak 4-compositions of size $N \geq 1$ could be used to model document collections of size N that were described in terms of these 4 parameters (discussed on page 112): the number of relevant documents that contained the query term, the number of relevant documents that did not contain the query term, the number of non-relevant documents that contained the query term, and the number of non-relevant documents that did not contain the query term. The next step in this process was the development of equations that were used at several places in this dissertation to calculate the quality of ranking (i.e., $\mathcal{Q}$) values for the coordination level matching (CLM), inverse document frequency (IDF) , and decision-theoretic (DT) ranking methods. Descriptions of the CLM, IDF, and DT ranking methods can be found in Sections 5.1, 6.1, and 6.3, respectively. These sections start on pages 144, 241, and 246, respectively.

There were several versions of the quality of ranking equations for the CLM, IDF, and DT ranking methods. The CLM versions are represented by Equation 5.0.1 (on page 143), Equation 5.9.3 (on page 196), Equation 5.10.3 (on page 205), and Equation 5.11.11 (on page 228). The IDF versions are represented by Equation 6.1.1 (on page 241), Equation 6.1.2 (on page 241), and Equation 6.1.3 (on page 242). Lastly, the DT versions are represented by Equation 6.3.1 (on page 247), Equation 6.3.6 (on page 250), and Equation 6.3.13 (on page 254).

These equations were developed in a manner such that they were well-defined even when singularities were present. Equations were also developed that could be used to calculate the expected value and variance of the $\mathcal{Q}$ values for the CLM (Equation 5.12 on page 228), IDF (Equation 6.1.1 on page 242), and DT (Equation 6.3.1 on page 254) ranking methods. Validation of the equations for these ranking methods occurred immediately after this step.

Validation mainly consisted of computing the value of a measure by both analytic and empirical means. In most cases, validation consisted of checking the two values to determine if they were exact matches (i.e., their values were equal). More specifically, validation consisted of these six activities: (1) using exact matching to compare the analytically-determined and empirically-determined ranking method-specific $\mathcal{Q}'$ values; (2) using the Wilcoxon signed ranks test to compare the $\mathcal{Q}'$ values that were estimated by random sampling with those values that were determined by analytic means; (3) using exact matching to compare the analytically-determined $\mathcal{A}'$ values, for $1 \leq N \leq 200$, with their empirically-determined counterparts, and for $201 \leq N \leq 400$, by using analytic means to generate boundary values that were checked to see if they satisfied predetermined boundary conditions; (4) using exact matching to compare the analytically-determined and empirically-determined ranking method-specific ASL$'$ values; (5) determining if analytically-determined ASL'_{r} (i.e., refined ASL$'$) values satisfied certain inequalities and by checking to determine if an analytically-determined value was an exact match for the corresponding empirically-determined value; and (6) using exact matching to compare the analytically-determined and empirically-determined ranking method-specific ASL'_{g} (i.e., the gold standard ASL$'$) values.

The next-to-last step in this process tested the hypothesis that there were no statistically significant differences between the analytically- and empirically-determined values for the ASL$'$ measure. The final step determined how well the ASL measured performance when it was compared to each of these performance measures: MZE, ESL, and MRR.

11.4 Problems Conducting the Research

The section discusses 5 major problems that the author encountered when he was conducting the research for this dissertation. The author was able to create solutions for each of these problems. The remainder of this section provides detailed information on each of the 5 problems.

The first problem concerned combinatorial explosion. This occurred when the number of documents in a collection became greater than 500. Combinatorial explosion and space-time limitations prevented exhaustive (i.e., brute force) validation of the counting equations for the quality of ranking measures that were associated with the CLM and IDF ranking methods. The solution was to only perform brute force validation on document collections where the cardinality did not exceed 500. In order to validate the equations that were developed in Chapters 5 and 6 for the coordination level matching (CLM), inverse document frequency (IDF), and decision-theoretic (DT) ranking methods, it took over 30 hours of elapsed time on this author's computer for document collections where the cardinalities started at 1, and continually increased by step size of 1, until the cardinality limit of 500 was reached.

The second problem concerned running out of computer memory. This space resource problem occurred with Mathematica® during the generation of a given number x of random weak 4-compositions for the validation work in Chapter 8 that involved the estimation of $\mathcal{Q}$ values for document collections that contained up to 10 billion documents. Mathematica's algorithm for a specified number of random weak compositions with $k = 4$ parts attempted to first generate the set of all the weak 4-compositions for the number N of documents in a collection. Then it would randomly select x of these weak compositions as the result. The problem was that the number of weak 4-compositions for N grows at a cubic rate (shown by Equation 3.4.1 on page 102). For even relatively small values of N, say $N = 1,000$, the cardinality of this set was such that there was not enough memory to create a set of

$$\tilde{C}_4(1000) = \binom{1000+3}{3} = 167,668,501$$

weak 4-compositions on the author's computer. To work around this, the author needed to find an algorithm that conceptually generated these random weak 4-compositions one at a time, and stopped after it had generated x distinct weak 4-compositions. The author's solution to this problem was to create Mathematica® implementations of the RANCOM (random composition of n into k parts) and RANKSB (random k-subset of an n-set) algorithms whose FORTRAN (Friedman and Koffman, 1977) implementations are detailed in Nijenhuis and Wilf (1978). The RANCOM algorithm invokes the RANKSB algorithm to do the vast majority of the overall computations. Note that the use of the term "composition" in Nijenhuis and Wilf (1978) is equivalent to our use of the term "weak composition."

The third problem was data-dependent and occurred during hypothesis testing for Research Question 2 (RQ2). More specifically, the Wilcoxon signed ranks test generated many N.A. (not available) results because, in the matched pairs version of this test in the statistical computing and graphics language and environment known as R, the algorithm for this statistical test eliminates any observations where the observed and actual value in a matched pair are equal. In the situation that this test was used for in Chapter 9, many of the sets of values consisted of pairs of observations where either the actual and predicted values for a pair were identical, or a very high percentage of the pairs had actual and predicted values that were identical. The author decided to report the results from the Kolmogorov-Smirnov (K-S) test, instead, because the K-S test was much more tolerant of matched pairs where the values were equal.

The fourth problem concerned the fact that neither of the ASL, MZE, MRR, and ESL performance measures could be used to calculate the value of the measure at an arbitrary point in a ranking *and* also calculate the correct value of the measure when the document collection was weakly-ordered. Of the 4 performance measures (i.e., ASL, MZE, MRR, ESL) that were used to help find the answer for Research Question 3 (RQ3), only the ESL measure was guaranteed to compute correct results when the document collection was weakly ordered. This was a major problem because the query-document model in this dissertation used binary relevance, and, instead of using the term occurrence frequency for a document, all that mattered was whether a query term was present or absent. If the query term was present in a document, its frequency was considered to be 1; otherwise, the frequency was considered to be 0. Effectively, both relevance and term frequency each had only two distinct values, namely, 0 and 1. The result of these choices was that the rankings that were produced by the best case, coordination level matching, decision-theoretic, inverse document frequency, random case, and worst case ranking methods were weakly ordered. This necessitated that three of the performance measures (i.e., ASL, RR, MZE), that this research was using needed to be adapted so that they were able to calculate correct values for rankings that had duplicate retrieval status values. Another problem was that in order to develop the answer for RQ3, all 4 of these measures were required to be able to compute their performance values at arbitrary points from the front of a vector V of ranked documents. The only definitions that satisfied this requirement were the ones for the precision and recall measures. These problems and other concerns were thoroughly discussed, and solved, in Section 10.5 (which starts on page 425).

The fifth, and last, problem concerned a typographical error that the author discovered in the McSherry and Najork (2008) article. The error occurred in the equation that was developed to calculate the value of the reciprocal rank performance measure, at an arbitrary point k in a vector V of ranked documents, for a weakly ordered collection of documents. The version of the equation that was given in the article was incorrect due to a typographical error. This problem was solved by developing a corrected version of the RR measure for this dissertation, and proving that it was correct.

11.5 Findings

This research demonstrated that it was possible to analytically calculate the Average Search Length for a document collection of size $N \geq 1$, and the 6 given ranking methods, by utilizing only 4 parameters and these three assumptions: all document distributions of size N were equally likely, relevance was binary, and a term was either present or absent in a document. These analytically-determined results were validated by empirical means and shown to have no significant differences between them and the results that were empirically-determined. The 6 ranking methods that were involved in this study were best-case, coordination level matching, inverse document frequency, decision-theoretic, random, and worst-case ranking.

The main contribution of this research was a set of equations that enabled researchers to assess or study the performance of various ranking algorithms by analytic prediction techniques (in contrast to having to set up various experiments) (Losee, 1995). These equations relied on just 4 parameters: the number of relevant documents that contained the query term, the number of relevant documents that did not contain the query term, the number of non-relevant documents that contained the query term, and the number of non-relevant documents that did not contain the query term. For all document collections, the sum of the values associated with these parameters was equal to the number of documents in the collection. Via analytic techniques and specific observations, the quality of ranking equation for decision-theoretic ranking $\mathcal{Q}'_{\mathrm{DT}}$ was found to be dependent on just a single parameter, namely N, the number of documents in a collection.

By setting up various scenarios, the equations developed in this research could be used to study how various ranking algorithms perform when entities such as the number of documents in a collection and the presence, or absence, of a query term in a set of relevant and non-relevant documents were manipulated. It was envisioned that the equations developed in this research could lead to a better understanding of some aspects of the document ranking process.

In particular, this research provided the ability to estimate $\mathcal{Q}$ (the quality of a ranking method) with as few as one parameter value in one particular case (i.e., decision-theoretic ranking); provided a way to compare the $\mathcal{Q}$ values for several ranking methods for an arbitrary document collection size or for an arbitrary range of document collection sizes; enabled the study of under what condition(s) the quality of each of the ranking methods (e.g., inverse document frequency, coordination level matching, decision-theoretic ranking) was inferior to, the same as, or superior to the other two; and also enabled the study of what impact, if any, the size of the document collection had on $\mathcal{Q}$ as the size approached infinity. The next few paragraphs detail the findings that occurred in several of the chapters.

Chapter 4. The research showed that it was not difficult to adapt the singularity-handling method (discussed in Section 4.3 which starts on page 117) that was proposed by Shaw (1995) so that it would work well in the query-document model, that was used in this dissertation, with respect to the computations of p', q', and t'.

Chapter 5. The work in this chapter showed that weak 4-compositions and the Principle of Inclusion-Exclusion (discussed in Section 5.8 which starts on page 159) were very effective in the development of the equations to calculate the $\mathcal{Q}'_{\mathrm{CLM}}$ measure. The work to develop these equations was involved, tedious, laborious, and demonstrated the desirability of using a computer algebra system, such as Mathematica$^{®}$, to assist with the many calculations.

Chapter 6. The equation for $\mathcal{Q}'_{\mathrm{IDF}}$ was found to be a simple extension of the one for $\mathcal{Q}'_{\mathrm{CLM}}$. Practically, there was no significant difference between CLM ranking and IDF ranking once the number of documents in a collection approached 40. For both of these ranking methods, the quality of ranking value approached 0.5 (the theoretical expected value for random ranking). The quality of ranking values for all collections of size $N \geq 1$, for the decision-theoretic ranking method, was found to be higher than those values with corresponding collection sizes for the CLM and IDF ranking methods. For $N \geq 50$, the $\mathcal{Q}'_{\text{DT}}$ values approached 1 (the theoretical expected value for best-case ranking). From about $N = 25$, and upwards, the mean and standard deviation of the of the $\mathcal{Q}'_{\text{CLM}}$ and $\mathcal{Q}'_{\text{IDF}}$ values were found to be approximately the same. Furthermore, these values were found to be approximately equal to 0.5 at $N = 50$, and upwards. The standard deviation of $\mathcal{Q}'_{\text{DT}}$ was found to monotonically decrease as the size N of the document collection increased; it approached 0 around $N = 200$.

Chapter 7. This chapter was notable for several developments: a combinatorial model for $\mathcal{A}$, the use of Gaussian polynomials to model search lengths, equations that can be used to determine the expected value and variance of normalized and unnormalized search lengths, and refined versions of the ASL measure. The research found that it was possible to develop a combinatoric equation for $\mathcal{A}$ that was equivalent to the probabilistic version. It was also found that Gaussian polynomials could be used to obtain distributional information on the sums of the positions of the relevant documents in an optimal ranking.

Chapter 8. It was found that $\mathcal{Q}$ values could be estimated very accurately and efficiently by random sampling, even when the margin of error for the sampling was as high as 0.05. It was found that there were no significant differences, at the 95% confidence level for a two-tailed test using the normal distribution, between the actual $\mathcal{Q}$ values and the estimated $\mathcal{Q}$ values. This was true whether the $\mathcal{Q}$ values were generated with a 0.01 or 0.05 margin of error.

Chapter 9. As a whole, the performance measures (i.e., $\text{ASL}', \text{ASL}'_{\text{r}}$) that estimate the Average Search Length and the performance measure (i.e., ASL'_{g}) that is calculated from a process that ranks documents and, then, calculates the Average Search Length from this empirical ranking data, were found to produce statistically significant different results. Overall, the gold standard ASL measure (i.e., $\mathrm{ASL}'_{\mathrm{g}}$) was found to produce the same results that would be obtained empirically by the process that was described earlier in this paragraph, and the refined version of the ASL measure (i.e., $\mathrm{ASL}'_{\mathrm{r}}$) was found to produce results that were in many cases not as accurate as those produced by the $\mathrm{ASL}'_{\mathrm{g}}$ performance measure. But, on many other occasions, dependent, of course, on the distribution of documents in the collection, the $\mathrm{ASL}'_{\mathrm{r}}$ performance measure produced results that matched those produced by the $\mathrm{ASL}'_{\mathrm{g}}$ measure. Similarly, the ASL' measure was often found to produce the same values as the $\mathrm{ASL}'_{\mathrm{r}}$ measure but, on other occasions, the values that it produced deviated more from those produced by the $\mathrm{ASL}'_{\mathrm{g}}$ measure than did the values that were produced by the $\mathrm{ASL}'_{\mathrm{r}}$ measure. These three performance measures were found to conform to this relationship:

$$|\mathrm{ASL}'_{\mathrm{r}} - \mathrm{ASL}'_{\mathrm{g}}| \leq |\mathrm{ASL}' - \mathrm{ASL}'_{\mathrm{g}}|.$$

Chapter 10. The results from this chapter showed that the ASL performance measure did not always totally agree, or totally disagree, with the MZE, MRR, and ESL measures on the relative rankings of a document collection. Rather, the agreement-disagreement plots contained multiple regions; some of these were regions where the ASL and the other measure agreed on the relative ranking of the documents, whereas there were other regions that illustrated where these measures disagreed on the relative rankings. The somewhat surprising finding, at least initially, was that in one of the examples that were constructed for Chapter 10, out of 18 plots only 5 of them were distinct. Further research showed that this was attributable to a combination of factors: the distribution of the documents, the characteristics of the 6 ranking methods that were used, binary relevance, and because the ranking algorithms considered only whether a term was present or absent in a document. If a given term was present, and it occurred, say thirty times in a document, it was treated the same as if the term had occurred just once in the document.

11.6 Implications and Recommendations

The document cut-off measures that were developed in this dissertation can be used to help study any vector V of ranked documents, at arbitrary document cut-off points, provided that (1) relevance is binary and (2) the following information can be determined from the ranked output: the equivalence classes and their relative sequence, the number of documents in each equivalence class, and the number of relevant documents that each class contains. These measures can be used even when the query-document model allows more than two possible distinct values for the term frequency component.

The ESL@$k(V, x)$, ASL@$k(V)$, MZE@$k(V)$, RR@$k(V)$ measures produce correct results even when the document collection is weakly-ordered. For the convenience of the reader, these measures, their associated defining equations, and the pages that these equations appear on are listed below.

Measure	Defining Equations and Locations
ESL@$k(V, x)$	Equation 10.6.7 on page 450.
ASL@$k(V)$	Equation 10.6.14 on page 459.
MZE@$k(V)$	Equation 10.6.15 on page 462.
RR@$k(V)$	Equation 10.6.16 on page 468 and
	Equation 10.6.27 on page 476.

These new measures are guaranteed to deliver results that are at least as accurate as the standard versions of many of these information retrieval (IR) measures, where the versions in the IR literature typically assume that the rankings are strongly-ordered. All of these measures can be used to help study Web ranking because they incorporate the notion of arbitrary document cut-off points.

The Q values for the CLM, IDF, and DT ranking methods changed very little proportionately after $N = 50$ (a miniscule number because a typical real-world document collection is almost always going to contain more than 50 documents). Therefore, we could calculate the $\mathcal{Q}$ value at, say, N=50, for a ranking method m, and pretend that this value was a constant for ranking method m in the same sense that 0 was the expected constant $\mathcal{Q}$ value for worst-case ranking, 0.5 was the constant for random ranking, and 1 was the constant for best-case ranking.

For all practical purposes, the $\mathcal{Q}$ values for the CLM and IDF ranking methods were the same. Therefore, if a study involved both the CLM and IDF ranking measures, for the same size document collection, we could calculate the $\mathcal{Q}$ value for one of these ranking measures and use it as the $\mathcal{Q}$ value for both of them. Analytic techniques for the determination of ranking method-specific $\mathcal{Q}$ values for the query-document model used in this dissertation produced results that were identical to those that could be obtained by empirical techniques, but much more efficiently, and at a much lower cost with respect to computational resources such as, for example, processor time, processor speed, disk space, disk speed, and memory.

The author recommends that any researcher who contemplates performing similar research to that which occurred for this dissertation be familiar with many of these topical areas: elementary number theory (Rosen, 2005), analytic combinatorics (Flajolet and Sedgwick, 2009), applied combinatorics (Tucker, 1980; Gross, 2008; Roberts and Tesman, 2009), enumerative combinatorics (Liu, 1968; Comtet, 1974; Goulden and Jackson, 1983; Stanley, 1997; Charalambides, 2002; Bóna, 2006; Aigner, 2007; Bóna, 2007), concrete and discrete mathematics (Graham et al., 1994; Knuth, 1997; Rosen, 1999; Rosen et al., 2000; Benjamin and Quinn, 2003; Larsen, 2007), basic hypergeometric series (Slater, 1966; Gasper and Rahman, 2004), the analysis of algorithms (Purdom and Brown, 1985; Sedgewick and Flajolet, 1996; Knuth, 1997; Dobrushkin, 2009), probability theory and mathematical statistics (Terrell, 1999; Williams, 2001; Rose and Smith, 2002; Walpole, 2002), nonparametric statistics (Conover, 1999), discrete distributions (Charalambides, 2005; Johnson et al., 2005), and differential and integral calculus (Berkey, 1984; Kosmala, 1998). In particular, generating functions (Lando, 2003; Wilf, 2006) should be an area of concentration, or emphasis, when studying enumerative combinatorics. Of course, the particular areas that the prospective researcher would need to be familiar with, and the depths of the familiarities, would greatly depend on that person's research question(s), and, hence, could vary from one combination of researcher and research study to another combination of researcher and research study. Additionally, some familiarity with combinatorial algorithms (Reingold et al., 1977; Nijenhuis and Wilf, 1978; Kreher and Stinson, 1999; Pemmaraju and Skiena, 2003; Knuth, 2005a,b, 2006) could prove to be very useful during the software implementation phase(s) of the research.

11.7 Future Research

This section details several possibilities for extending this research:

1. the extension of the query-document model to handle multiple term queries;
2. the extension of the query-document model so that relevance remains discrete, but it can have more than two distinct values;
3. the extension of the query-document model so that relevance is continuous;
4. the extension of the query-document model to use actual term frequencies;
5. the elimination of the uniformity assumption, that is, no longer assuming that each weak 4-composition in the set of weak 4-compositions for a document collection of size N is as equally likely to be chosen as any other member of this set; and
6. the application of this research to distributed information retrieval performance contexts.

Item 1 represents the most natural extension of the research in this dissertation, it allows for multiple term queries. But, the possibility that queries may have more than one term means that the query-document model may need to be enhanced to also incorporate information on term dependencies, unless the model assumes that the terms are independent. This independence assumption is a common simplifying assumption in the IR literature for studies that involve multiple term queries (Losee, 1998; Metzler and Croft, 2005).

Item 2 extends the query-document model so that relevance can have more than two distinct values. Relevance is still discrete, but it is no longer dichotomous. That is, there would be different degrees of relevance (Tang et al., 1999; Kekäläinen and Järvelin, 2002).

Item 3 extends the query-document model to handle continuous relevance. Losee (1998) proposes a way to incorporate continuous relevance into an analytic model of text filtering. This way can also be used for information retrieval. The implication of using the Losee proposal to extend this dissertation research is that the author would, most likely, need to switch from the discrete mathematical techniques that he used for this dissertation research and, instead, would need to switch to continuous mathematical techniques and integral calculus.

Item 4 concerns the extension of the query-document model so that it could handle actual term frequencies, as contrasted with frequencies that were conflated into just two values (i.e., 1, if the query term is present in the document; 0, if the query term was absent). The Losee (1998) article states that term frequencies can be incorporated into an IR model if these frequencies are considered to be Poisson-distributed (Harter, 1975a,b; Bookstein, 1983; Raghavan et al., 1983; Losee et al., 1986; Srinivasan, 1990; Fuhr, 1992; Margulis, 1993; Robertson and Walker, 1994; Ponte and Croft, 1998; Robertson, 2004; Lee and Lee, 2005). The article goes on to state that even though a Poisson distribution assumption may not exactly model a particular natural language situation, the accuracy of the model is good enough that it can be used effectively for information retrieval.

Item 5 is an extension that allows the model to weight some sets of weak 4-compositions for a document collection of size N different than some of the other sets for this collection. This enables the modeling of situations where document collections with some characteristics are more, or less, likely to occur than others. If, say, the weight for a weak 4-composition was a value in the closed interval $[0, 1]$, then a weight of 0 could be given to those weak 4-compositions that, via prior knowledge, are known not to occur, even though an instance of them is theoretically possible, if one did not have this prior knowledge.

Item 6 refers to the applicability of this research to the measuring of IR performance in distributed information retrieval contexts. Losee and Church (2004) developed analytical techniques for predicting distributed information retrieval performance in various collection fusion scenarios for both uniprocessor and multiprocessor scenarios. The research in this dissertation can be used to extend the research on the problems that were studied in the Losee and Church (2004) article.

11.8 Summary

This research investigated the characteristics of analytic performance measures for studying and predicting the performance of IR systems and of systems that have both information retrieval and database capabilities. It used these performance measures for *prediction*, rather than mainly for retrospection, which is quite different from how many IR performance measures have been used in the past. These predictive measures were used, in lieu of empirical techniques, to study the Average Search Length performance measure for the best case, coordination level matching, decision-theoretic, inverse document frequency, random case, and worst case ranking methods. The salient feature of this research was the formulation of a theory, with respect to information retrieval performance measures and document ranking methods, that centered mainly around composition theory, partition theory, and enumerative combinatorics; the development of a model for this theory; and being able to empirically validate the theoretical results that this model was expected to produce.

Based on the work for this dissertation, the following observations can be made: (1) this research enabled the modeling of ranking methods and performance measures by the use of enumerative combinatorics and concepts from number theory, calculus, set theory, probability theory, statistics, and discrete mathematics; (2) the analytic results from the equations that were developed for the quality of ranking methods, and the IR performance measures, matched the expected results which were obtained empirically by brute force (i.e., exhaustive) techniques; and (3) the extension of the ASL, ESL, MZE, and RR performance measures, so that performance could be calculated at arbitrary points in a ranking, and that also calculate the correct results for weakly-ordered document collections, open more opportunities for the use of these measures, particularly in situations (e.g., Web search) where all of the documents in a collection are typically not returned to the user.

Appendices

Appendix A

Creating the Modified Cystic Fibrosis Test Collection

A.1 Create the CF$'$ test collection

A.1.1 Transform the queries

Build the set of new queries by visiting each of the original queries and eliminating any stopwords from it. The remaining terms, for each query, after stemming, constitute the terms of the new version of that query and are the only differences between it and the original query. The symbol $\leftarrow$ denotes assigning the value on its right hand side to the variable on its left hand side.

```
$Q_{\mathrm{CF}'} \leftarrow$ *the empty set*

for each query q $\in Q_{\mathrm{CF}}$
   query_id $\leftarrow$ access$(q, 0)$
   the_bag_of_stemmed_terms $\leftarrow$ *the empty bag*

   for each term t $\in$ access$(q, 1)$
     if t is not a PubMed stopword
       the_stemmed_term $\leftarrow$ Porter_stemmer(t)
       insert the_stemmed_term into the_bag_of_stemmed_terms
     endif
   endfor

   if the_bag_of_stemmed_terms $\neq$ *the empty bag*
     insert <query_id, the_bag_of_stemmed_terms> into $Q_{\mathrm{CF}'}$
   endif

endfor
```

A.1.2 Transform the documents

Build the set of new documents by visiting each of the original documents and eliminating any stopwords from it. The remaining terms, for each document, after stemming, constitute the terms of the new version of that document and are the only differences between it and the original document.

```
$D_{\mathrm{CF}'} \leftarrow$ *the empty set*

for each document d $\in D_{\mathrm{CF}}$
   document_id $\leftarrow$ access$(d, 0)$
   the_bag_of_stemmed_terms $\leftarrow$ *the empty bag*

   for each term t $\in$ access$(d, 1)$
     if t is not a PubMed stopword
       the_stemmed_term $\leftarrow$ Porter_stemmer(t)
       insert the_stemmed_term into the_bag_of_stemmed_terms
     endif
   endfor

   if the_bag_of_stemmed_terms $\neq$ *the empty bag*
     insert <document_id, the_bag_of_stemmed_terms> into $D_{\mathrm{CF}'}$
   endif

endfor
```

A.1.3 Transform the relevance judgments

Build the set of new relevance judgment associations by visiting each of the original associations and mapping the 4 relevance judgments there into a single Y (relevant) or N (not-relevant) judgment. The only difference between it and the original association is that 4 items have been mapped into just 1 item.

```
$J_{\mathrm{CF}'} \leftarrow$ the empty set

for each triple <query_id,document_id,rj> $\in J_{\mathrm{CF}}$
  score $\leftarrow$ $\mathrm{access(rj, 1) + access(rj, 2) + access(rj, 3) + access(rj, 1)}$
  if score $\geq$ 1
    insert <query_id,document_id,Y> into $J_{\mathrm{CF}'}$
  else
    insert <query_id,document_id,N> into $J_{\mathrm{CF}'}$
  endif
endfor
```

A.2 Select the best single term description of each query in the CF′ test collection

In order to select the best single term, we need to perform the following actions for each query q in the CF′ test collection. For each query q, let z be the query identifier for it.

Compute the set of document identifiers for the documents that are relevant to q.

$\mathrm{docids}_q \leftarrow \{\ did \mid <qid, did, rj> \in J_{\mathrm{CF}'}$ and $z = qid$ and $rj = Y\ \}$.

Compute the relevance set for q.

$\mathrm{relset}_q \leftarrow \{\ <did, doc> \mid <did, doc> \in D_{\mathrm{CF}'}$ and $did \in docids_q\ \}$.

Define a language model for the relevance set. The language model for relset_q is calculated by first concatenating all the documents in it to form a single large document (shown by Equation A.2.1 on the next page). Let this combined document be represented by R where $d_1, d_2, d_3, ..., d_{|relset_q|}$ are documents from relset_q.

$$R \leftarrow d_1 \oplus d_2 \oplus d_3 \oplus ... \oplus d_{|\mathrm{relset}_q|} \tag{A.2.1}$$

Remember that each element of $relset_q$ is a pair whose second component is a document represented as a bag of terms. Informally, we define the effect of the concatenation operation (i.e., $\oplus$) on two documents as follows: it unions the bags of terms, preserving duplicates. That is, if a term t occurs in either of the documents, it must also appear in the result of the concatenation. Also, only terms that are in at least one of the documents are eligible for membership in the bag that results from the concatenation operation. Furthermore, the number of occurrences in the result for a term t is the sum of the number of times that it appears in both documents — if t occurs $n_1 \geq 0$ times in one document and $n_2 \geq 0$ times in the other one, then it occurs $n_1 + n_2$ times in the result.

The language model can then be determined easily using standard methods described in Section 2.7 – we estimate the probability of each term by the use of Equation A.2.2 and we apply smoothing by the use of Equation A.2.3,

$$\widehat{P}_{\mathrm{mle}}(t|M_R) = \frac{tf_{t,M_R}}{dl_R} \tag{A.2.2}$$

$$P_{\mathrm{jm}}(t|M_R) = \lambda \widehat{P}_{\mathrm{mle}}(t|M_R) + (1-\lambda)\widehat{P}_{\mathrm{mle}}(t|M_{\mathrm{corpus}}) \tag{A.2.3}$$

$\widehat{P}_{\mathrm{mle}}(t|M_R)$ and $\widehat{P}_{\mathrm{mle}}(t|\mathrm{corpus})$ are the maximum likelihood estimates for R and the corpus, respectively. $P_{\mathrm{jm}}(t|R)$ is the Jelinek-Mercer smoothing method.

The weight that is applied to $\widehat{P}_{\mathrm{mle}}(t|M_R)$ is $\lambda = 0.6$ in order to be consistent with the value used in Lavrenko and Croft (2001), Cronen-Townsend and Croft (2002), and Jordan *et al.* (2006).

Define a language model for the corpus. The corpus model is estimated from all the documents in the collection. Since this model contains all of the documents, rather than a subset of them, it is considered to be rather complete with respect to its term population. Therefore, we assume that the maximum likelihood estimator adequately approximates it. Hence, no smoothing is applied to the corpus language model.

$$\text{corpus} \leftarrow d_1 \oplus d_2 \oplus d_3 \oplus ... \oplus d_{|D_{CF'}|}$$

where $d_1, d_2, d_3, ..., d_{|D_{CF'}|}$ are documents from $D_{CF'}$.

$$\widehat{P}_{\text{mle}}(t|M_{\text{corpus}}) = \frac{\text{tf}_{t,M_{\text{corpus}}}}{dl_{\text{corpus}}}$$

The maximum likelihood estimator above for a term t that occurs in the corpus is calculated by counting how many times it occurs in the corpus divided by the total number of terms in the corpus.

Calculate the contribution that each term t in the vocabulary V makes to the relative entropy of the two language models (i.e., M_R and M_{corpus}). Terms that contribute the least to relative entropy can be viewed as the terms that least distinguish the relevance set from the corpus. Terms that contribute the most are those that most distinguish the relevance set from the corpus. The calculation for this term discrimination value appears immediately below.

$$\text{term\_discrimination\_value}(t) = P_{\text{jm}}(t|M_R) \log \frac{P_{\text{jm}}(t|M_R)}{\widehat{P}_{\text{mle}}(t|M_{\text{corpus}})}$$

This approach is similar to Cai *et al.* (2001) who used this scoring function to find terms for query expansion and to Jordan *et al.* (2006) who used this function to automatically synthesize queries of varying degrees of quality in their study of blind relevance feedback.

Sort the terms in V based on how much they contribute to relative entropy.

Identify the sorted term that contributes the most to relative entropy. This term is the one in V that most distinguishes those in the relevance set for the query from those in the corpus. It is the one that has just now been chosen to represent the single term version of q and is denoted by q'.

Appendix B

Turning multiple term queries into single term queries

It is important to state that the sole purpose of this example is to illustrate a method for distilling multiple term queries into a single term query. That is all that it does. Unfortunately, it takes many pages and a fair amount of calculations to do so. Its importance to the overall work associated with this dissertation is that it shows how we create the single term queries that form the query portion of our test collection.

B.1 Example

Assume that we have the six short "documents" below, numbered from 1 to 6, inclusively; that their associated language models are named M_1, M_2, ..., M_6, respectively; that punctuation marks are treated as delimiters; and that the case of the words in the documents is insignificant. For this example, we make matters easier to understand by choosing not to do term normalization (i.e., stemming and stopword elimination are not performed). The language models for the documents are in Table B.1 on page 524. Texts for the six documents follow.

1. The ability to distinguish between acceptable and unacceptable levels of retrieval performance and the ability to distinguish between significant and non-significant differences between retrieval results are important to traditional information retrieval experiments.

Burgin (1999)

2. Discusses issues of diversity in library and information science-education programs and how these efforts can be addressed positively to better serve students and their future users. Topics include a historical background, attracting people of diversity for doctoral programs and faculty positions, curriculum issues, and recruiting.

   Gollop (1999)

3. This paper reports on the automatic metadata generation applications (AMeGA) project's metadata expert survey. Automatic metadata generation research is reviewed and the study's methods, key findings and conclusions are presented.

   Greenberg, Spurgin, and Crystal (2006)

4. Probabilistic document retrieval systems consistent with the 2-Poisson independence model outperforms the binary independence model if the terms are distributed as described by the model's assumptions.

   Losee (1986)

5. Information theory is concerned with the transmission of information, through a channel, to a receiver. The sender and receiver could be people or machines. In most cases they are different, but when information is being stored for later retrieval, the receiver could be the sender at some future time.

   Luenberger (2006)

6. Alternatively, the idea of information seeking in context offers encouragement to loosen the structures of terminology, research foci, methods, and assumptions about ideal behavior to discover what the role of information in people's lives is.

   Solomon (1999)

Let us further assume that we are interested in computing the probability that each of the documents generated the same particular query q. To figure that out, we need to estimate the probability of producing the language model M_d of document d using maximum likelihood estimation (MLE), given the bag of words assumption. We accomplish this by using MLE to compute the probability of each query term t for language model M_d and then multiplying these individual probabilities to obtain the joint probability. The probability that a specific term t occurs in a specific document d is estimated by determining how many times t occurs in d, then dividing that quantity by the number of terms in d. Equation 2.7.3 on page 52 succinctly expresses what has been discussed in this paragraph.

We can use Document 1 to illustrate how we obtain these probabilities. That document has 33 terms (according to our parsing rules), 21 of which are unique. In the language model associated with that document, the probability for *ability* is $\frac{2}{33}$ because that term occurs 2 times out of 33 in that document; the probability for *acceptable* is $\frac{1}{33}$ because that term occurs 1 time out of 33; and the probability for *and* is $\frac{3}{33}$ because that term occurs 3 times in the document. We can use the same technique to calculate the probabilities of each unique remaining term appearing in this document. The same technique can be applied to the terms in the other documents. These document language model-specific probabilities are listed in Table B.1 on the following page; those for the corpus are listed in Table B.2 on page 525. Following that, Table B.3 on page 526 lists the probability of each of the query q terms t (i.e., *information*, *retrieval*, *performance*) for each of the 6 language models.

Using Equation 2.7.3 on page 52 and the data in Table B.3 on page 526, we can estimate the probability that each of our 6 language models produced the query q. These calculations are detailed in Table B.4 on page 526.

Table B.1: The Unigram Language Models for the Documents

Model M_1		Model M_2		Model M_3		Model M_4		Model M_5		Model M_6	
ability	$\frac{2}{33}$	a	$\frac{1}{46}$	amega	$\frac{1}{32}$	are	$\frac{1}{28}$	a	$\frac{2}{49}$	about	$\frac{1}{36}$
acceptable	$\frac{1}{33}$	addressed	$\frac{1}{46}$	and	$\frac{2}{32}$	as	$\frac{1}{28}$	and	$\frac{1}{49}$	alternatively	$\frac{1}{36}$
and	$\frac{3}{33}$	and	$\frac{5}{46}$	applications	$\frac{1}{32}$	assumptions	$\frac{1}{28}$	are	$\frac{1}{49}$	and	$\frac{1}{36}$
are	$\frac{1}{33}$	attracting	$\frac{1}{46}$	are	$\frac{1}{32}$	binary	$\frac{1}{28}$	at	$\frac{1}{49}$	assumptions	$\frac{1}{36}$
between	$\frac{3}{33}$	background	$\frac{1}{46}$	automatic	$\frac{2}{32}$	by	$\frac{1}{28}$	be	$\frac{2}{49}$	behavior	$\frac{1}{36}$
differences	$\frac{1}{33}$	be	$\frac{1}{46}$	conclusions	$\frac{1}{32}$	consistent	$\frac{1}{28}$	being	$\frac{1}{49}$	context	$\frac{1}{36}$
distinguish	$\frac{2}{33}$	better	$\frac{1}{46}$	expert	$\frac{1}{32}$	described	$\frac{1}{28}$	but	$\frac{1}{49}$	discover	$\frac{1}{36}$
experiments	$\frac{1}{33}$	can	$\frac{1}{46}$	findings	$\frac{1}{32}$	distributed	$\frac{1}{28}$	cases	$\frac{1}{49}$	encouragement	$\frac{1}{36}$
important	$\frac{1}{33}$	curriculum	$\frac{1}{46}$	generation	$\frac{2}{32}$	document	$\frac{1}{28}$	channel	$\frac{1}{49}$	foci	$\frac{1}{36}$
information	$\frac{1}{33}$	discusses	$\frac{1}{46}$	is	$\frac{1}{32}$	if	$\frac{1}{28}$	concerned	$\frac{1}{49}$	idea	$\frac{1}{36}$
levels	$\frac{1}{33}$	diversity	$\frac{2}{46}$	key	$\frac{1}{32}$	independence	$\frac{2}{28}$	could	$\frac{2}{49}$	ideal	$\frac{1}{36}$
non	$\frac{1}{33}$	doctoral	$\frac{1}{46}$	metadata	$\frac{3}{32}$	model	$\frac{3}{28}$	different	$\frac{1}{49}$	in	$\frac{2}{36}$
of	$\frac{1}{33}$	education	$\frac{1}{46}$	methods	$\frac{1}{32}$	outperforms	$\frac{1}{28}$	for	$\frac{1}{49}$	information	$\frac{2}{36}$
performance	$\frac{1}{33}$	efforts	$\frac{1}{46}$	on	$\frac{1}{32}$	poisson	$\frac{1}{28}$	future	$\frac{1}{49}$	is	$\frac{1}{36}$
results	$\frac{1}{33}$	faculty	$\frac{1}{46}$	paper	$\frac{1}{32}$	probabilistic	$\frac{1}{28}$	in	$\frac{1}{49}$	lives	$\frac{1}{36}$
retrieval	$\frac{3}{33}$	for	$\frac{1}{46}$	presented	$\frac{1}{32}$	retrieval	$\frac{1}{28}$	information	$\frac{3}{49}$	loosen	$\frac{1}{36}$
significant	$\frac{2}{33}$	future	$\frac{1}{46}$	project	$\frac{1}{32}$	s	$\frac{1}{28}$	is	$\frac{2}{49}$	methods	$\frac{1}{36}$
the	$\frac{2}{33}$	historical	$\frac{1}{46}$	reports	$\frac{1}{32}$	systems	$\frac{1}{28}$	later	$\frac{1}{49}$	of	$\frac{3}{36}$
to	$\frac{3}{33}$	how	$\frac{1}{46}$	research	$\frac{1}{32}$	terms	$\frac{1}{28}$	machines	$\frac{1}{49}$	offers	$\frac{1}{36}$
traditional	$\frac{1}{33}$	in	$\frac{1}{46}$	reviewed	$\frac{1}{32}$	the	$\frac{4}{28}$	most	$\frac{1}{49}$	people	$\frac{1}{36}$
unacceptable	$\frac{1}{33}$	include	$\frac{1}{46}$	s	$\frac{2}{32}$	two	$\frac{1}{28}$	of	$\frac{1}{49}$	research	$\frac{1}{36}$
		information	$\frac{1}{46}$	study	$\frac{1}{32}$	with	$\frac{1}{28}$	or	$\frac{1}{49}$	role	$\frac{1}{36}$
		issues	$\frac{2}{46}$	survey	$\frac{1}{32}$			people	$\frac{1}{49}$	s	$\frac{1}{36}$
		library	$\frac{1}{46}$	the	$\frac{2}{32}$			receiver	$\frac{3}{49}$	seeking	$\frac{1}{36}$
		of	$\frac{2}{46}$	this	$\frac{1}{32}$			retrieval	$\frac{1}{49}$	structures	$\frac{1}{36}$
		people	$\frac{1}{46}$					sender	$\frac{2}{49}$	terminology	$\frac{1}{36}$
		positions	$\frac{1}{46}$					some	$\frac{1}{49}$	the	$\frac{3}{36}$
		positively	$\frac{1}{46}$					stored	$\frac{1}{49}$	to	$\frac{2}{36}$
		programs	$\frac{2}{46}$					the	$\frac{4}{49}$	what	$\frac{1}{36}$
		recruiting	$\frac{1}{46}$					theory	$\frac{1}{49}$		
		science	$\frac{1}{46}$					they	$\frac{1}{49}$		
		serve	$\frac{1}{46}$					through	$\frac{1}{49}$		
		students	$\frac{1}{46}$					time	$\frac{1}{49}$		
		their	$\frac{1}{46}$					to	$\frac{1}{49}$		
		these	$\frac{1}{46}$					transmission	$\frac{1}{49}$		
		to	$\frac{1}{46}$					when	$\frac{1}{49}$		
		topics	$\frac{1}{46}$					with	$\frac{1}{49}$		
		users	$\frac{1}{46}$								

Table B.2: The Unigram Language Model for the Corpus

a	$\frac{3}{224}$	concerned	$\frac{1}{224}$	historical	$\frac{1}{224}$	or	$\frac{1}{224}$	stored	$\frac{1}{224}$
ability	$\frac{2}{224}$	conclusions	$\frac{1}{224}$	how	$\frac{1}{224}$	outperforms	$\frac{1}{224}$	structures	$\frac{1}{224}$
about	$\frac{1}{224}$	consistent	$\frac{1}{224}$	idea	$\frac{1}{224}$	paper	$\frac{1}{224}$	students	$\frac{1}{224}$
acceptable	$\frac{1}{224}$	context	$\frac{1}{224}$	ideal	$\frac{1}{224}$	people	$\frac{3}{224}$	study	$\frac{1}{224}$
addressed	$\frac{1}{224}$	could	$\frac{2}{224}$	if	$\frac{1}{224}$	performance	$\frac{1}{224}$	survey	$\frac{1}{224}$
alternatively	$\frac{1}{224}$	curriculum	$\frac{1}{224}$	important	$\frac{1}{224}$	poisson	$\frac{1}{224}$	systems	$\frac{1}{224}$
amega	$\frac{1}{224}$	described	$\frac{1}{224}$	in	$\frac{4}{224}$	positions	$\frac{1}{224}$	terminology	$\frac{1}{224}$
and	$\frac{12}{224}$	differences	$\frac{1}{224}$	include	$\frac{1}{224}$	positively	$\frac{1}{224}$	terms	$\frac{1}{224}$
applications	$\frac{1}{224}$	different	$\frac{1}{224}$	independence	$\frac{2}{224}$	presented	$\frac{1}{224}$	the	$\frac{15}{224}$
are	$\frac{4}{224}$	discover	$\frac{1}{224}$	information	$\frac{7}{224}$	probabilistic	$\frac{1}{224}$	their	$\frac{1}{224}$
as	$\frac{1}{224}$	discusses	$\frac{1}{224}$	is	$\frac{4}{224}$	programs	$\frac{2}{224}$	theory	$\frac{1}{224}$
assumptions	$\frac{2}{224}$	distinguish	$\frac{2}{224}$	issues	$\frac{2}{224}$	project	$\frac{1}{224}$	these	$\frac{1}{224}$
at	$\frac{1}{224}$	distributed	$\frac{1}{224}$	key	$\frac{1}{224}$	receiver	$\frac{3}{224}$	they	$\frac{1}{224}$
attracting	$\frac{1}{224}$	diversity	$\frac{2}{224}$	later	$\frac{1}{224}$	recruiting	$\frac{1}{224}$	this	$\frac{1}{224}$
automatic	$\frac{2}{224}$	doctoral	$\frac{1}{224}$	levels	$\frac{1}{224}$	reports	$\frac{1}{224}$	through	$\frac{1}{224}$
background	$\frac{1}{224}$	document	$\frac{1}{224}$	library	$\frac{1}{224}$	research	$\frac{2}{224}$	time	$\frac{1}{224}$
be	$\frac{3}{224}$	education	$\frac{1}{224}$	lives	$\frac{1}{224}$	results	$\frac{1}{224}$	to	$\frac{7}{224}$
behavior	$\frac{1}{224}$	efforts	$\frac{1}{224}$	loosen	$\frac{1}{224}$	retrieval	$\frac{5}{224}$	topics	$\frac{1}{224}$
being	$\frac{1}{224}$	encouragement	$\frac{1}{224}$	machines	$\frac{1}{224}$	reviewed	$\frac{1}{224}$	traditional	$\frac{1}{224}$
better	$\frac{1}{224}$	experiments	$\frac{1}{224}$	metadata	$\frac{3}{224}$	role	$\frac{1}{224}$	transmission	$\frac{1}{224}$
between	$\frac{3}{224}$	expert	$\frac{1}{224}$	methods	$\frac{2}{224}$	s	$\frac{4}{224}$	two	$\frac{1}{224}$
binary	$\frac{1}{224}$	faculty	$\frac{1}{224}$	model	$\frac{3}{224}$	science	$\frac{1}{224}$	unacceptable	$\frac{1}{224}$
but	$\frac{1}{224}$	findings	$\frac{1}{224}$	most	$\frac{1}{224}$	seeking	$\frac{1}{224}$	users	$\frac{1}{224}$
by	$\frac{1}{224}$	foci	$\frac{1}{224}$	non	$\frac{1}{224}$	sender	$\frac{2}{224}$	what	$\frac{1}{224}$
can	$\frac{1}{224}$	for	$\frac{2}{224}$	of	$\frac{7}{224}$	serve	$\frac{1}{224}$	when	$\frac{1}{224}$
cases	$\frac{1}{224}$	future	$\frac{2}{224}$	offers	$\frac{1}{224}$	significant	$\frac{2}{224}$	with	$\frac{2}{224}$
channel	$\frac{1}{224}$	generation	$\frac{2}{224}$	on	$\frac{1}{224}$	some	$\frac{1}{224}$		

Table B.3: Document Term Probabilities for Query q (before smoothing)

q	information	retrieval	performance
M_1	$\frac{1}{33}$	$\frac{3}{33}$	$\frac{1}{33}$
M_2	$\frac{1}{46}$	0	0
M_3	0	0	0
M_4	0	$\frac{1}{28}$	0
M_5	$\frac{3}{49}$	$\frac{1}{49}$	0
M_6	$\frac{2}{36}$	0	0

Table B.4: Estimated Probabilities for Query q (before smoothing)

$$\widehat{P}(q|M_1) = \tfrac{1}{33} \cdot \tfrac{3}{33} \cdot \tfrac{1}{33} = \tfrac{3}{35937}$$
$$\widehat{P}(q|M_2) = \tfrac{1}{46} \cdot 0 \cdot 0 = 0$$
$$\widehat{P}(q|M_3) = 0 \cdot 0 \cdot 0 = 0$$
$$\widehat{P}(q|M_4) = 0 \cdot \tfrac{1}{28} \cdot 0 = 0$$
$$\widehat{P}(q|M_5) = \tfrac{3}{49} \cdot \tfrac{1}{49} \cdot 0 = 0$$
$$\widehat{P}(q|M_6) = \tfrac{2}{36} \cdot 0 \cdot 0 = 0$$

We find that, except for model M_1, these calculated probabilities are 0 because each of the other models is missing at least one of the query terms. Table B.3 on the preceding page illustrates this; these are the probabilities that we obtain without smoothing. For the reasons stated in the quote above from Manning *et al.* (2008), it is considered good practice to work with smoothed (rather than non-smoothed) probabilities.

The formula for calculating smoothed probabilities is below.

$$\widehat{P}(w|d) = \lambda \widehat{P}_{\text{mle}}(w|M_d) + (1 - \lambda)\widehat{P}_{\text{mle}}(w|M_c)$$

The weight that is applied to $\widehat{P}_{mle}(w|M_d)$ is $\lambda = 0.6$, in order to be consistent with the value used in Lavrenko and Croft (2001), Cronen-Townsend and Croft (2002), and Jordan *et al.* (2006). Higher values of λ are more suitable for short queries, lower values are more suitable for long queries (Manning et al., 2008).

Smoothing ensures that any term that appears in the document collection has a non-zero probability. The main justification for this "is that a non-occurring term is possible in a query, but no more likely than would be expected by chance from the whole collection" (Manning et al., 2008). How do we accomplish this? Basically, we use the concept of a finite mixture distribution (McLachlan and Peel, 2000) to compute the smoothed probabilities. In our example, this means that the smoothed probability for a term in a document is a linear combination of the maximum likelihood estimation probabilities of the document term and the corresponding corpus term. Since, by definition, a corpus contains all of the terms in the documents that comprise it, the probability of a corpus term is always non-zero. Also, because $0 < \lambda < 1$, the smoothed probability for a term in a document typically has a value different than the corresponding non-smoothed probability. Conceptually, the effect of this is to add to the language model for a document, the terms that appear in the corpus but not in the document. Initially, these added terms have a zero probability. The smoothing process can be viewed as sharing the wealth (i.e., probability mass) of a document among those terms that were added to the document's language model but originally only appeared in at least one of the other documents. Basically, it is redistributing the probability mass so that all terms (including the added ones – with their initial zero probabilities) of a language model have a nonzero probability. This is important because the language model is a probability mass function (i.e., the probabilities of its terms must always sum to 1 and must still be a probability mass function after the smoothing has occurred. The smoothing process preserves this property of a language model. This is also important later when we use relative entropy to calculate how dissimilar two language models are. A key requirement of the relative entropy calculation is that its two parameters represent probability mass functions. If they do not, then the value generated by the calculation may lack validity.

Smoothing transformed the information in Table B.3 on page 526 to that in Table B.5. Utilization of the information in the latter table transformed the information in Table B.4 on page 526 to that in Table B.6 on the following page. Before discussing the information in the Table B.6 on the next page and its relevance to some of what this research is attempting to do, it would be a very good idea to show how the information in Table B.5 was derived from that in Table B.3 on page 526.

Table B.5: Document Term Probabilities for Query q (after smoothing)

q	information	retrieval	performance
M_1	$\frac{27}{880}$	$\frac{391}{6160}$	$\frac{123}{6160}$
M_2	$\frac{47}{1840}$	$\frac{1}{112}$	$\frac{1}{560}$
M_3	$\frac{1}{80}$	$\frac{1}{112}$	$\frac{1}{560}$
M_4	$\frac{1}{80}$	$\frac{17}{560}$	$\frac{1}{560}$
M_5	$\frac{193}{3920}$	$\frac{83}{3920}$	$\frac{1}{560}$
M_6	$\frac{11}{240}$	$\frac{1}{112}$	$\frac{1}{560}$

First, we created the corpus by combining the 6 documents into a single document. This yielded a document with 224 terms, of which 134 were unique. Table B.2 on page 525

Table B.6: Estimated Probabilities for Query q (after smoothing)

$$
\begin{aligned}
\widehat{P}(q|M_1) &= \tfrac{27}{880} \cdot \tfrac{3191}{6160} \cdot \tfrac{123}{6160} = 3.88867 \text{ x } 10^{-5} \\
\widehat{P}(q|M_2) &= \tfrac{47}{1840} \cdot \tfrac{1}{112} \cdot \tfrac{1}{560} = 4.072620896184561 \text{ x } 10^{-7} \\
\widehat{P}(q|M_3) &= \tfrac{1}{80} \cdot \tfrac{1}{112} \cdot \tfrac{1}{560} = 1.9929846938775508 \text{ x } 10^{-7} \\
\widehat{P}(q|M_4) &= \tfrac{1}{80} \cdot \tfrac{17}{560} \cdot \tfrac{1}{560} = 6.776147959183674 \text{ x } 10^{-7} \\
\widehat{P}(q|M_5) &= \tfrac{193}{3920} \cdot \tfrac{83}{3920} \cdot \tfrac{1}{560} = 1.8615522921996787 \text{ x } 10^{-6} \\
\widehat{P}(q|M_6) &= \tfrac{11}{240} \cdot \tfrac{1}{112} \cdot \tfrac{1}{560} = 7.307610544217686 \text{ x } 10^{-7}
\end{aligned}
$$

represents the unigram language model M_{corpus} for this corpus. Next, we applied the Jelinek-Mercer smoothing method to compute the smoothed probabilities for each combination of document language model and query term. Finally, we used the results of that to replace each probability in Table B.3 on page 526 with its corresponding smoothed probability.

To illustrate, let us compute the smoothed probability for the *retrieval* term in language model M_1. The way that we calculate this is similar to the way that we determine it for all the other document language model/query term combinations.

$$
\begin{aligned}
\widehat{P}(\text{retrieval}|M_1) &= \lambda \widehat{P}_{\text{mle}}(\text{retrieval}|M_1) + (1-\lambda)\widehat{P}_{\text{mle}}(\text{retrieval}|M_{\text{corpus}}) \\
&= \frac{6}{10} \cdot \frac{3}{33} + \left(1 - \frac{6}{10}\right) \frac{5}{224} \\
&= \frac{6}{10} \cdot \frac{3}{33} + \frac{4}{10} \cdot \frac{5}{224} \\
&= \frac{18}{330} + \frac{20}{2240} \\
&= \frac{18}{330} \cdot \frac{224}{224} + \frac{20}{2240} \cdot \frac{33}{33} \\
&= \frac{4032}{73920} + \frac{660}{73920} \\
&= \frac{4032 + 660}{73920} \\
&= \frac{4692}{73920}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{12 \cdot 391}{12 \cdot 6160} \\
&= \frac{\cancel{12} \cdot 391}{\cancel{12} \cdot 6160} \\
&= \frac{391}{6160}
\end{aligned}
$$

If we sort the estimated probabilities in Table B.6 on the preceding page, from the highest to the lowest, we find that $\widehat{P}(q|M_1) > \widehat{P}(q|M_5) > \widehat{P}(q|M_6) > \widehat{P}(q|M_4) > \widehat{P}(q|M_2) > \widehat{P}(q|M_3)$. This means that Document 1 is the most likely document to have produced the query. This does not exclude the possibility that any of the other 5 documents, however, could have produced the query. All it says is that it is most likely that it was produced by Document 1. Extra confidence in this result comes from the fact that the estimated probability for Document 1 is greater than that for Document 5 by approximately an order of magnitude and that Document 1 was the only document that contained all three of the query terms.

So far, we have shown how the unigram language model can be used to generate estimated probabilities that can be used to rank a collection of documents according to how likely they were to have produced a query q. However, that is not our main interest in using these models. What we are much more interested in is in determining the best single term to represent a multiple term query. This is pivotal to the research being performed in this investigation. Without a way to both do that effectively and having a very good theoretical basis for doing so, this research could not take place. Section 3.2.1 provides an algorithm for selecting the best single term for a multiple term query. We use the 6 documents described in this Appendix to provide more detail as to how the algorithm works.

The first two steps of the algorithm are concerned with finding out what the query q is and then identifying the set of documents that are relevant to q. Next, we define a unigram language model M_R for the relevance set. A relevance set is simply the group of documents that are known to be relevant to q. These documents are concatenated to form a single, possibly large, document. From this single document, the unigram language model for the relevance set is constructed. Following this, we construct a language model for the corpus (this is the concatenation of all of the documents in the collection to form a single, possibly large, document). Now that we have a language model for both the relevance set and the corpus, we can smooth the probabilities in the former language model so that terms that appear in the corpus, but not in the query, have non-zero probabilities. The way to do that was discussed earlier in this example.

Suppose that query q consists of just the following three terms – information, retrieval, research – and that the relevance set for this query has only two documents, namely, Document 1 and Document 4. Also, let the corpus be the same six document corpora that we used earlier in this example. Table B.7 lists the terms in the non-smoothed language model for the relevance set – which has 61 terms, with 40 of them being unique.

Table B.7: The Unigram Language Model for the Relevance Set

ability	$\frac{2}{61}$	by	$\frac{1}{61}$	if	$\frac{1}{61}$	outperforms	$\frac{1}{61}$	systems	$\frac{1}{61}$
acceptable	$\frac{1}{61}$	consistent	$\frac{1}{61}$	important	$\frac{1}{61}$	performance	$\frac{1}{61}$	terms	$\frac{1}{61}$
and	$\frac{3}{61}$	described	$\frac{1}{61}$	independence	$\frac{2}{61}$	poisson	$\frac{1}{61}$	the	$\frac{6}{61}$
are	$\frac{2}{61}$	differences	$\frac{1}{61}$	information	$\frac{1}{61}$	probabilistic	$\frac{1}{61}$	to	$\frac{3}{61}$
as	$\frac{1}{61}$	distinguish	$\frac{2}{61}$	levels	$\frac{1}{61}$	results	$\frac{1}{61}$	traditional	$\frac{1}{61}$
assumptions	$\frac{1}{61}$	distributed	$\frac{1}{61}$	model	$\frac{3}{61}$	retrieval	$\frac{4}{61}$	two	$\frac{1}{61}$
between	$\frac{3}{61}$	document	$\frac{1}{61}$	non	$\frac{1}{61}$	s	$\frac{1}{61}$	unacceptable	$\frac{1}{61}$
binary	$\frac{1}{61}$	experiments	$\frac{1}{61}$	of	$\frac{1}{61}$	significant	$\frac{2}{61}$	with	$\frac{1}{61}$

The next action that we have to perform is to calculate the contribution that each term t in the vocabulary V (i.e., the terms in the corpus) makes to the relative entropy of the two language models (i.e., M_R and M_{corpus}).

Terms that contribute the least to relative entropy can be viewed as the terms that least distinguish the relevance set from the corpus. Terms that contribute the most are those that most distinguish the relevance set from the corpus. The function to compute the discrimination value for a term t is below.

$$\text{term\_discrimination\_value}(t) = P_{\text{jm}}(t|M_R) \log \frac{P_{\text{jm}}(t|M_R)}{\widehat{P}_{\text{mle}}(t|M_{\text{corpus}})}$$

Using term_discrimination_value(t), we compute the discrimination power of each term t, then sort the terms in V based on how much they contribute to relative entropy. From those terms, we identify the sorted term that contributes the most to relative entropy. This term is the one in V that most distinguishes those in the relevance set for the query from those in the corpus. Below is how we calculate the discrimination value for the term *retrieval*. First, we calculate the smoothed value for the term, then we use that value as one of the inputs to the equation for the term discrimination value.

$$\begin{aligned}
P_{\text{jm}}(\text{retrieval}|M_R) &= \lambda \widehat{P}_{\text{mle}}(\text{retrieval}|M_R) + (1-\lambda)\widehat{P}_{\text{mle}}(\text{retrieval}|M_{\text{corpus}}) \\
&= \frac{6}{10} \cdot \frac{4}{61} + \left(1 - \frac{6}{10}\right) \frac{5}{224} \\
&= \frac{6}{10} \cdot \frac{4}{61} + \frac{4}{10} \cdot \frac{5}{224} \\
&= \frac{24}{610} + \frac{20}{2240} \\
&= \frac{24}{610} \cdot \frac{224}{224} + \frac{20}{2240} \cdot \frac{61}{61} \\
&= \frac{5376}{136640} + \frac{1220}{136640} \\
&= \frac{5376 + 1220}{136640} \\
&= \frac{6596}{136640} \\
&= \frac{4 \cdot 1649}{4 \cdot 34160} \\
&= \frac{\cancel{4} \cdot 1649}{\cancel{4} \cdot 34160} \\
&= \frac{1649}{34160}
\end{aligned}$$

$$\begin{aligned}
\text{term\_discrimination\_value(retrieval)} &= P_{\text{jm}}(\text{retrieval}|M_R)\log\frac{P_{\text{jm}}(\text{retrieval}|M_R)}{\widehat{P}_{\text{mle}}(\text{retrieval}|M_{\text{corpus}})}\\
&= \frac{1649}{34160}\log\frac{1649/34160}{5/224}\\
&= \frac{1649}{34160}\log\left(\frac{1649}{34160}\cdot\frac{224}{5}\right)\\
&= \frac{1649}{34160}\log\frac{369376}{170800}\\
&= \frac{1649}{34160}\log\frac{112\cdot 3298}{112\cdot 1525}\\
&= \frac{1649}{34160}\log\frac{\cancel{112}\cdot 3298}{\cancel{112}\cdot 1525}\\
&= \frac{1649}{34160}\log\frac{3298}{1525}\\
&= 0.0482728\log 2.16262\\
&= 0.0482728\times 0.771322\\
&= 0.0372339
\end{aligned}$$

Table B.8 on the following page lists the 9 terms that contribute the most and the least to the relative entropy between the documents in the relevance set and those in the corpus. It can be readily seen that the most discriminating term is *retrieval* and that there is a tie between *of* and *information* for the least discriminating term. This means that the best single term query for the documents in the relevance set is one that has the sole term *retrieval*.

Table B.8: The Nine Most Discriminating and the Nine Least Discriminating Terms

Most Discriminating		Least Discriminating	
term	discrimination value	term	discrimination value
retrieval	0.0372339	receiver	-0.00490870
model	0.0333582	people	-0.00490870
between	0.0333582	metadata	-0.00490870
significant	0.0222388	be	-0.00490870
independence	0.0222388	a	-0.00490870
distinguish	0.0222388	is	-0.00654493
ability	0.0222388	in	-0.00654493
the	0.0212690	of	-0.00750082
to	0.0124279	information	-0.00750082

Appendix C

The Derivation of A Formula to Calculate the Expected Position of a Specified Relevant Document in An Equivalence Class

Lemma C.0.1. *Suppose* $1 \leq i \leq r \leq n$ *and* $i, r, n, l \in \mathbb{N}$. *Let* $[l, l+n-1]$ *represent positions* $l, l+1, \ldots, l+n-1$ *in an equivalence class of* n *documents with exactly* r *relevant documents. Assuming that a relevant document has the same probability of occupying any one of these* n *positions as it does of occupying any one of the other* $n-1$ *positions, the expected mean position for the* i*th relevant document from the beginning of the interval is*

$$i - 1 + l + i(n-r)/(r+1).$$

Proof. The are $\binom{n}{r}$ distinct sequences of documents that are associated with the n positions in the closed interval $[l, l+n-1]$. Each sequence has r relevant documents and $m = n-r$ non-relevant documents. For an arbitrary sequence, the ith relevant document in it partitions the sequence into three parts. The first part is the *prefix* and contains $i-1$ relevant documents and $0 \leq m \leq n-r$ non-relevant documents. These documents can be arranged in any of

$$\binom{i-1+m}{i-1} = \binom{i-1+m}{m}$$

orders. The second part of an arbitrary sequence consists of just a single document, namely, the ith relevant document, which is at the location that corresponds to position

$$l + (i-1) + m = l + i - 1 + m$$

in the sequence. The third part of the sequence is the *suffix* and consists of the remaining

$$n - (i+m)$$

documents; $r-i$ of these documents are relevant and the remaining $(n-r)-m$ documents are non-relevant. The documents in the suffix can be arranged independently of those in the first 2 parts of the sequence. The number of such distinct orders is

$$\binom{n-(i+m)}{r-i} = \binom{n-i-m)}{n-r-m}.$$

Figure C.1 on the next page depicts the relationships that we have just described in this paragraph.

From the information in the above paragraph, and assuming that each distinct sequence is equally likely, the expected position of the ith relevant document x_{EPIRD}, over all the possible sequences, can be determined by this equation:

$$\begin{aligned} x_{\text{EPIRD}} &= \binom{n}{r}^{-1} \sum_{m=0}^{n-r} (m+i+l-1) \binom{m+i-1}{m} \binom{n-m-i}{n-r-m} \\ &= \binom{n}{r}^{-1} (A+B) \qquad \text{(C.0.1)} \end{aligned}$$

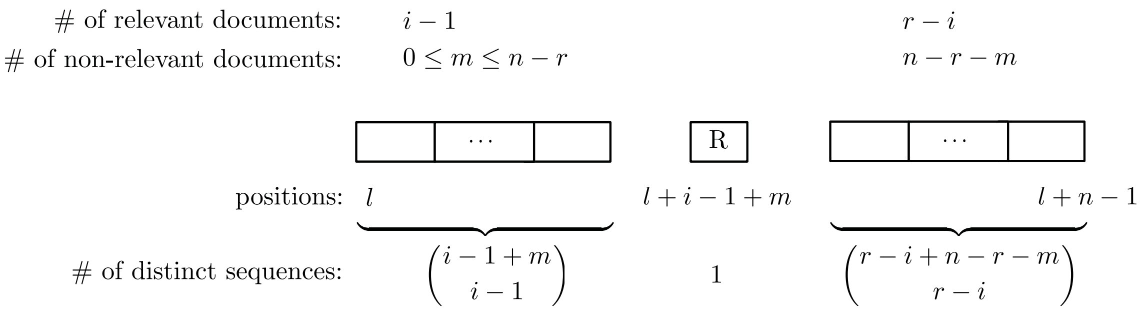


Figure C.1: This diagram details the basic relationships that are associated with the documents in the equivalence class.

where the equations for A and B are detailed below.

As usual, our goal is to reduce an equation, such as this one, to a closed form, if possible. To make progress towards this goal, we attempt to find closed form expressions for A and B, and then use these expressions to rewrite Equation C.0.1 on the preceding page.

$$
\begin{aligned}
A &= \sum_{m=0}^{n-r}(i+l-1)\binom{m+i-1}{m}\binom{n-m-i}{n-r-m} \\
&= (i+l-1)\sum_{m=0}^{n-r}\binom{m+i-1}{m}\binom{n-m-i}{n-r-m} \\
&= (i+l-1)\sum_{m=0}^{n-r}\binom{m+i-1}{i-1}\binom{n-m-i}{r-i} \\
&= (i+l-1)\binom{n}{r}. \qquad \text{(C.0.2)} \\
B &= \sum_{m=0}^{n-r} m\binom{m+i-1}{m}\binom{n-m-i}{n-r-m}. \qquad \text{(C.0.3)}
\end{aligned}
$$

By algebraic and combinatorial manipulations, the expression

$$m\binom{m+i-1}{m},$$

in Equation C.0.3 on the previous page, can be simplified in this way:

$$\begin{aligned}
m\binom{m+i-1}{m} &= m\frac{((i-1)+m)!}{m!(i-1)!} && \text{(by Equation 10.4.5 on page 422)}\\
&= \frac{((i-1)+m)!}{(m-1)!(i-1)!} && \text{(by dividing numerator and denominator by } m\text{)}\\
&= i\frac{((i-1)+m)!}{(m-1)!i!} && \text{(by multiplying numerator and denominator by } i\text{)}\\
&= i\binom{(i-1)+m}{m-1} && \text{(by Equation 10.4.5 on page 422)}\\
&= i\binom{(i-1)+m}{i}. && \text{(by Equation 10.4.6 on page 422)}
\end{aligned}$$

By the use of this simplification, we can now derive a closed form version of Equation C.0.3 on page 537:

$$\begin{aligned}
B &= \sum_{m=0}^{n-r} m\binom{m+i-1}{m}\binom{n-m-i}{n-r-m}\\
&= \sum_{m=0}^{n-r} i\binom{(i-1)+m}{i}\binom{n-m-i}{n-r-m}\\
&= i\sum_{m=0}^{n-r} \binom{(i-1)+m}{i}\binom{n-m-i}{n-r-m}\\
&= i\sum_{m=0}^{n-r} \binom{(i-1)+m}{i}\binom{n-m-i}{r-i}\\
&= i\binom{n}{r+1}. \qquad \text{(C.0.4)}
\end{aligned}$$

At this point, we have closed form versions of the equations for A and B. We can use this information, along with that from Equation C.0.1 on page 536, to express x_{EPIRD} as a closed form equation. This can be accomplished by the use of Equation C.0.2 on the previous page and Equation C.0.4. They enable us to rewrite Equation C.0.1 on page 536 as

$$
\begin{aligned}
x_{\text{EPIRD}} &= \binom{n}{r}^{-1} \sum_{m=0}^{r} (m+i+l-1) \binom{m+i-1}{m} \binom{n-m-i}{n-r-m} \\
&= \binom{n}{r}^{-1} (A+B) \\
&= \binom{n}{r}^{-1} \left((i+l-1) \binom{n}{r} + i \binom{n}{r+1} \right) \\
&= (i+l-1) + i \binom{n}{r+1} \binom{n}{r}^{-1} \\
&= (i+l-1) + i(n-r)/(r+1)
\end{aligned}
$$

because

$$
\begin{aligned}
\binom{n}{r+1} \binom{n}{r}^{-1} &= \frac{n!}{(r+1)!(n-r-1)!} \frac{r!(n-r)!}{n!} \\
&= \frac{r!(n-r)!}{(r+1)!(n-r-1)!} \\
&= \frac{r!(n-r)(n-r-1)!}{(r+1)r!(n-r-1)!} \\
&= (n-r)/(r+1).
\end{aligned}
$$

This completes the proof. □

Appendix D

Derivation of the Alternate Equation for $\mathcal{Q}$ for the IDF Ranking Method

The *document term weight* (DTW) w_d for the inverse document frequency (IDF) ranking method, as stated in Losee (1998), for a document that contains the desired feature (e.g., term), that is, it has feature frequency 1 is

$$w_d = -\log t.$$

If this document does not contain the feature, that is, it has feature frequency 0, then its DTW is

$$w_d = 0.$$

The *query term weight* (QTW) w_q for a query that contains the feature is

$$w_q = 1.$$

If the query does not contain the feature, its QTW is

$$w_q = 0.$$

The retrieval status value (RSV) for a given query-document pair is a function of the QTW for the query and the DTW for the document. Its RSV is calculated as

$$\mathrm{RSV}_{q,d} = w_q \cdot w_d$$

for the single term query model that is being used in this dissertation.

The idea behind each of the ranking methods in this dissertation is to rank documents that contain the term ahead of documents that do not contain the term. For a given query-document pair with query term weight w_q and document term weight w_d, the equation for the ranking value that is assigned to that document is effectively

$$\mathrm{RSV}_{q,d} = \begin{cases} w_d, & \text{if } w_q = 1; \\ 0, & \text{otherwise.} \end{cases}$$

The case that is of interest in developing the equation for $\mathcal{Q}_{\mathrm{CLM}}$ is those instances in which $w_d = 1$. In such an instance, the weight

$$w_d = -\log t$$

increases as t approaches 1. However, when t reaches 1 the weight decreases to 0. Since logarithms are undefined when their argument is 0, this implies that documents with a feature frequency of 1 are always ranked ahead of documents with a feature frequency of 0 for the IDF ranking method when the joint conditions

$$w_q = 1 \text{ and } 0 < p < 1$$

hold. Documents with a feature frequency of 1 are ranked the same as, or lower than, documents with feature frequency 0 only when $-\log(t) \leq 0$. This situation can only occur when $t = 1$ holds. The discussion in this paragraph leads to the following sequence of derivations:

$$\begin{aligned}
\mathcal{Q}_{\text{IDF}}(p,t) &= \Pr(p > t, t > 0) + \Pr(p \leq t, t \leq 0) && \text{(D.0.1)}\\
&= \Pr(p > t, t > 0) + \Pr(p \leq t, t = 1) && \text{(D.0.2)}\\
&= \Pr(p > t) + \Pr(p \leq t, t = 1) && \text{(D.0.3)}
\end{aligned}$$

In Chapter 4, it was mentioned that p is undefined for some of the weak 4-compositions that correspond to some of the queries that can occur in a document collection of size N. In that chapter, the issue of singularities and some techniques to handle them were discussed. The result of the discussions there was alternate definitions for p, t, and q that were able to gracefully handle singularities in the various contexts that these entities were being used in.

The initial attempt at the modified equation for the quality of the IDF ranking method yields

$$\mathcal{Q}_{\text{IDF}}(p',t') = \Pr(p' > t') + \Pr(p' \leq t', t' = 1).$$

This equation is correct except for the $t' = 1$ part. The problem is that 1 is the maximum value for t but it is not the maximum value for t'. In fact, the maximum value for t' is slightly less than the maximum value for its counterpart due to the singularity-handling technique that was chosen in Chapter 4. Based on this technique,

$$\text{the maximum value of } t' = \begin{cases} 1 - N^{-2}, & \text{if } N \geq 2;\\ 1 - 10^{-4}, & \text{otherwise.} \end{cases}$$

The corrected equation is

$$\mathcal{Q}_{\text{IDF}}(p',t') = \Pr(p' > t') + \Pr(p' \leq t', t' = 1 - \epsilon),$$

where

$$\epsilon = \begin{cases} N^{-2}, & \text{if } N \geq 2; \\ 10^{-4}, & \text{otherwise.} \end{cases}$$

Appendix E

The Number of Qualifying Weak 4-Compositions for Selected Ranking Methods

Table E.1: Number of Qualifying Contributions ($1 \leq N \leq 40$)

N	nqc_CLM	nqc_IDF	nqc_DT	N	nqc_CLM	nqc_IDF	nqc_DT
1	0	2	3	21	958	980	2003
2	1	4	8	22	1096	1119	2278
3	4	8	17	23	1254	1278	2577
4	9	14	31	24	1388	1413	2901
5	18	24	51	25	1580	1606	3251
6	28	35	78	26	1763	1790	3628
7	46	54	113	27	1962	1990	4033
8	64	73	157	28	2167	2196	4467
9	90	100	211	29	2422	2452	4931
10	119	130	276	30	2630	2661	5426
11	160	172	353	31	2930	2962	5953
12	195	208	443	32	3184	3217	6513
13	254	268	547	33	3484	3518	7107
14	306	321	666	34	3801	3836	7736
15	370	386	801	35	4124	4160	8401
16	444	461	953	36	4449	4486	9103
17	536	554	1123	37	4866	4904	9843
18	615	634	1312	38	5236	5275	10622
19	732	752	1521	39	5638	5678	11441
20	829	850	1751	40	6040	6081	12301

Table E.2: Number of Qualifying Contributions ($41 \leq N \leq 120$)

N	nqc_CLM	nqc_IDF	nqc_DT	N	nqc_CLM	nqc_IDF	nqc_DT
41	6540	6582	13203	81	47412	47494	95203
42	6955	6998	14148	82	49181	49264	98688
43	7504	7548	15137	83	51004	51088	102257
44	7979	8024	16171	84	52653	52738	105911
45	8508	8554	17251	85	54634	54720	109651
46	9098	9145	18378	86	56568	56655	113478
47	9706	9754	19553	87	58510	58598	117393
48	10244	10293	20777	88	60444	60533	121397
49	10934	10984	22051	89	62612	62702	125491
50	11565	11616	23376	90	64509	64600	129676
51	12268	12320	24753	91	66768	66860	133953
52	12965	13018	26183	92	68935	69028	138323
53	13754	13808	27667	93	71194	71288	142787
54	14454	14509	29206	94	73486	73581	147346
55	15278	15334	30801	95	75786	75882	152001
56	16068	16125	32453	96	78048	78145	156753
57	16960	17018	34163	97	80656	80754	161603
58	17851	17910	35932	98	83027	83126	166552
59	18792	18852	37761	99	85530	85630	171601
60	19615	19676	39651	100	88065	88166	176751
61	20710	20772	41603	101	90850	90952	182003
62	21686	21749	43618	102	93380	93483	187358
63	22680	22744	45697	103	96254	96358	192817
64	23760	23825	47841	104	98888	98993	198381
65	24880	24946	50051	105	101680	101786	204051
66	25973	26040	52328	106	104703	104810	209828
67	27236	27304	54673	107	107696	107804	215713
68	28377	28446	57087	108	110475	110584	221707
69	29638	29708	59571	109	113742	113852	227811
70	30852	30923	62126	110	116674	116785	234026
71	32270	32342	64753	111	119938	120050	240353
72	33480	33553	67453	112	123028	123141	246793
73	35004	35078	70227	113	126504	126618	253347
74	36391	36466	73076	114	129673	129788	260016
75	37800	37876	76001	115	133140	133256	266801
76	39315	39392	79003	116	136565	136682	273703
77	40866	40944	82083	117	140040	140158	280723
78	42394	42473	85242	118	143696	143815	287862
79	44122	44202	88481	119	147286	147406	295121
80	45644	45725	91801	120	150740	150861	302501

Table E.3: Number of Qualifying Contributions ($121 \leq N \leq 200$)

N	nqc_CLM	nqc_IDF	nqc_DT	N	nqc_CLM	nqc_IDF	nqc_DT
121	154770	154892	310003	161	360428	360590	721603
122	158571	158694	317628	162	366957	367120	734968
123	162424	162548	325377	163	374004	374168	748497
124	166319	166444	333251	164	380689	380854	762191
125	170350	170476	341251	165	387470	387636	776051
126	174216	174343	349378	166	394708	394875	790078
127	178626	178754	357633	167	401886	402054	804273
128	182656	182785	366017	168	408584	408753	818637
129	186988	187118	374531	169	416260	416430	833171
130	191185	191316	383176	170	423407	423578	847876
131	195780	195912	391953	171	430902	431074	862753
132	199945	200078	400863	172	438475	438648	877803
133	204646	204780	409907	173	446254	446428	893027
134	209276	209411	419086	174	453698	453873	908426
135	213768	213904	428401	175	461490	461666	924001
136	218528	218665	437853	176	469284	469461	939753
137	223516	223654	447443	177	477460	477638	955683
138	228179	228318	457172	178	485541	485720	971792
139	233312	233452	467041	179	493772	493952	988081
140	237987	238128	477051	180	501429	501610	1004551
141	243298	243440	487203	181	510330	510512	1021203
142	248466	248609	497498	182	518440	518623	1038038
143	253634	253778	507937	183	527134	527318	1055057
144	258684	258829	518521	184	535588	535773	1072261
145	264296	264442	529251	185	544404	544590	1089651
146	269773	269920	540128	186	553063	553250	1107228
147	275170	275318	551153	187	562056	562244	1124993
148	280797	280946	562327	188	571007	571196	1142947
149	286602	286752	573651	189	579924	580114	1161091
150	292000	292151	585126	190	589118	589309	1179426
151	298150	298302	596753	191	598690	598882	1197953
152	303820	303973	608533	192	607600	607793	1216673
153	309810	309964	620467	193	617504	617698	1235587
154	315791	315946	632556	194	626961	627156	1254696
155	322048	322204	644801	195	636340	636536	1274001
156	328023	328180	657203	196	646121	646318	1293503
157	334646	334804	669763	197	656306	656504	1313203
158	340926	341085	682482	198	665790	665989	1333102
159	347338	347498	695361	199	676302	676502	1353201
160	353616	353777	708401	200	686000	686201	1373501

Bibliography

Aigner, M. (2007). *A Course in Enumeration*, Volume 238 of *Graduate Texts in Mathematics*. Berlin: Springer.

Andrews, G. E. (1974, October). Applications of basic hypergeometric functions. *SIAM Review 16*(4), 441–484.

Andrews, G. E. (1984). *The Theory of Partitions*. Cambridge [Cambridgeshire]; New York, NY, USA: Cambridge University Press.

Andrews, G. E. (1986). *q-series: Their Development and Application in Analysis, Number Theory, Combinatorics, Physics, and Computer Algebra*. Number 66 in CBMS Regional Conference Series in Mathematics. Providence, RI: American Mathematical Society.

Andrews, G. E. and K. Eriksson (2004). *Integer Partitions*. Cambridge, UK ; New York: Cambridge University Press.

Apostol, T. M. (1967). *Calculus, Vol. 1: One-Variable Calculus with an Introduction to Linear Algebra* (2 ed.). Waltham, Mass.: Blaisdell Pub. Co.

Arce, G. R. and M. Tian (1996). Order statistic filter banks. *IEEE Transactions on Image Processing 5*(6), 827–837.

Bachman, C. W. (1969). Data structure diagrams. *SIGMIS Database 1*(2), 4–10.

Baeza-Yates, R. and B. Ribeiro-Neto (1999, May). *Modern Information Retrieval*. Addison Wesley.

Ballerini, J., M. Büchel, R. Domenig, D. Knaus, B. Mateev, E. Mittendorf, P. Schäuble, P. Sheridan, and M. Wechsler (1996). SPIDER retrieval system at TREC–5. In *TREC-5 Proceedings*.

Barton, D. E. (1959). Review: [untitled]. *Journal of the Royal Statistical Society. Series A (General) 122*(1), 102–103.

Batini, C., S. Ceri, and S. B. Navathe (1992). *Conceptual Database Design: An Entity-Relationship Approach.* Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc.

Belkin, N. J., R. N. Oddy, and H. M. Brooks (1982). ASK For Information Retrieval: Part I. Background and Theory. *Journal of Documentation 38*(2), 61–71.

Benjamin, A. and J. J. Quinn (2003). *Proofs That Really Count: The Art of Combinatorial Proof*, Volume Dolciani mathematical expositions ; no. 27. [Washington, DC]: Mathematical Association of America.

Berger, A. and J. Lafferty (1999). Information retrieval as statistical translation. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 222–229. ACM.

Berkey, D. D. (1984). *Calculus.* Philadelphia: Saunders College Pub.

Berndt, B. C. and K. Ono (2001). *Q-Series With Applications to Combinatorics, Number Theory, and Physics : A Conference on q-series With Applications to Combinatorics, Number Theory, and Physics, October 26-28, 2000, University of Illinois.* Providence, R.I.: American Mathematical Society.

Berners-Lee, T. and M. Fischetti (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web.* San Francisco, CA, USA: Harper.

Blanco, R. and F. Silvestri (2008). ECIR 2008 Workshop on Efficiency Issues on Information Retrieval. *SIGIR Forum 42*(1), 59–62.

Bloch, E. D. (2000). *Proofs and Fundamentals: A First Course in Abstract Mathematics.* Boston: Birkhäuser.

Blumenfeld, D. (2001). *Operations Research Calculations Handbook.* Boca Raton: CRC Press.

Bollmann, P. and V. S. Cherniavsky (1981). Measurement-theoretical investigation of the mz-metric. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, Kent, UK, UK, pp. 256–267. Butterworth & Co.

Bóna, M. (2006). *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory* (2nd ed.). New Jersey: World Scientific.

Bóna, M. (2007). *Introduction to Enumerative Combinatorics.* Boston: McGraw-Hill Higher Education.

Bookstein, A. (1983, September). Information retrieval: A sequential learning process. *American Society for Information Science 34*(5), 331–342.

Borlund, P. (2003). The concept of relevance in IR. *Journal of the American Society for Information Science and Technology 54*(10), 913–925.

Browne, R. H. (2001, November). Using the sample range as a basis for calculating sample size in power calculations. *The American Statistician 55*(4), 293–298.

Bruce, T. (1992). *Designing Quality Databases with IDEF1X Information Models.* Dorset House.

Buckley, C. and E. M. Voorhees (2005). Retrieval system evaluation. In E. Voorhees and D. K. Harman (Eds.), *TREC: experiment and evaluation in information retrieval*, Digital libraries and electronic publishing, Chapter 3, pp. 53–75. Cambridge, Mass.: MIT Press.

Burgin, R. (1999). The Monte Carlo method and the evaluation of retrieval system performance. *Journal of the American Society for Information Science 50*(2), 181–191.

Cai, D. and C. J. van Rijsbergen (2004). A case study for automatic query expansion based on divergence. Technical report, University of Glasgow, Department of Computing Science.

Cai, D., C. J. van Rijsbergen, and J. M. Jose (2001). Automatic query expansion based on divergence. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, New York, NY, USA, pp. 419–426. ACM Press.

Chambers, J. M. (2008). *Software for Data Analysis: Programming with R.* New York: Springer.

Charalambides, C. A. (2002). *Enumerative Combinatorics*. Boca Raton: Chapman & Hall/CRC.

Charalambides, C. A. (2005). *Combinatorial Methods in Discrete Distributions*. Wiley series in probability and statistics. Hoboken, N.J: Wiley-Interscience.

Chaudhuri, S., R. Ramakrishnan, and G. Weikum (2005). Integrating DB and IR technologies: What is the sound of one hand clapping? In *CIDR*, pp. 1–12.

Chen, P. P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems 1*(1), 9–36.

Chiu, J. L.-T., R. T. Lin, H.-J. Dai, and R. T.-H. Tsai (2008). Improving the performance and stability of question answering system's accuracy with new feature and evaluation measurement. In *Proceedings of the ICDC'08 2008 International Conference on Digital Content.*, Chungli, Taiwan.

Cleverdon, C. (1997). The Cranfield tests on index language devices. In K. Sparck-Jones and P. Willett (Eds.), *Readings in Information Retrieval*, pp. 47–59. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Comtet, L. (1974). *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. Dordrecht, Boston: D. Reidel Publishing Company.

Conover, W. J. (1999). *Practical Nonparametric Statistics*. New York: Wiley.

Cooper, M. D. (1971a). *Evaluation of Information Retrieval Systems: A Simulation and Cost Approach*. Ph. D. thesis, School of Librarianship, University of California-Berkeley, Berkeley, Calif. 94720.

Cooper, W. (1968). Expected Search Length: A Single Measure of Retrieval Effectiveness Based on the Weak Ordering Action of Retrieval Systems. *American Documentation 19*, 30–41.

Cooper, W. S. (1967). Mathematical supplement to expected search length: A single measure of retrieval expectedness based on the weak ordering action of retrieval systems. Xeroxed copy.

Cooper, W. S. (1971b). A definition of relevance for information retrieval. *Information*

Storage and Retrieval 7, 19–37.

Cooper, W. S. (1973). On selecting a measure of retrieval effectiveness, part I: The 'subjective' philosophy of evaluation. *Journal of the American Society for Information Science 24*(2), 87–100.

Cover, T. M. and J. A. Thomas (2006). *Elements of Information Theory*. Hoboken, N.J: Wiley-Interscience.

Croft, W. and D. Harper (1979). Using probabilistic models of information retrieval without relevance information. *Journal of Documentation 35*, 285–295.

Cronen-Townsend, S., Y. Zhou, and W. B. Croft (2002). Predicting query performance. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 299–306. ACM Press.

Cuadra, C. A. and R. V. Katter (1967). Experimental studies of relevance judgments. Technical report, Systems Development Corporation, Santa Monica, CA.

Dalgaard, P. (2008). *Introductory Statistics with R* (2nd ed.). Springer.

David, F. N. (1959). Review: [untitled]. *Biometrika 46*(1/2), 271.

de Vries, A. P. and T. Roelleke (2005). Relevance information: A loss of entropy but a gain for IDF? In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 282–289. ACM.

Dobrushkin, V. A. (2009). *Methods in Algorithmic Analysis*. Boca Raton, Fla.: Chapman & Hall/CRC.

Dominich, S. (2001). *Mathematical Foundations of Information Retrieval*. Number 12 in Mathematical modelling–theory and applications. Dordrecht; Boston: Kluwer Academic Publishers.

Dong, L. and C. Watters (2004). Improving efficiency and relevance ranking in information retrieval. In *WI '04: Proceedings of the Web Intelligence, IEEE/WIC/ACM*

International Conference on (WI'04), Washington, DC, USA, pp. 648–651. IEEE Computer Society.

Downie, J. S., K. West, A. Ehmann, and E. Vincent (2005). The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR), 2005.*

Fine, N. J. (1988). *Basic Hypergeometric Series and Applications.* American Mathematical Society.

Flajolet, P. and R. Sedgwick (2009). *Analytic Combinatorics.* New York: Cambridge University Press.

Flanagan, D. (2005). *Java in a Nutshell.* Beijing ; Sebastopol, CA: O'Reilly.

Fox, C. (1992). Lexical analysis and stoplists. In *Information Retrieval: Data Structures and Algorithms*, Chapter 7, pp. 102–130. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Fox, E. A. (1983). Characterization of two new experimental collections in computer and information science containing textual and bibliographic concepts. Technical Report TR83-561, Department of Computer Science, Cornell University, Ithaca, NY.

Frakes, W. B. and R. Baeza-Yates (1992). *Information Retrieval: Data Structures and Algorithms.* Englewood Cliffs, N.J: Prentice Hall.

Frakes, W. B. and C. J. Fox (2003). Strength and similarity of affix removal stemming algorithms. *SIGIR Forum 37*(1), 26–30.

Friedman, F. L. and E. B. Koffman (1977). *Problem Solving and Structured Programming in FORTRAN.* Reading, Mass.: Addison-Wesley Publishing Company.

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal 35*(3), 243–255.

Gasper, G. and M. Rahman (2004). *Basic Hypergeometric Series*, Volume Encyclopedia of mathematics and its applications ; v. 96. Cambridge, UK ; New York: Cambridge University Press.

Gessel, I. (1985, April). Review: Combinatorial enumeration. *Bulletin (New Series) of the American Mathematical Society 12*(2), 297–301.

Gollop, C. J. (1999, July). Library and information science education: Preparing librarians for a multicultural society. *College & Research Libraries 60*(4), 385–395.

Goulden, I. P. and D. M. Jackson (1983). *Combinatorial Enumeration.* Somerset, New Jersey: John Wiley & Sons, Inc.

Graham, R. L., D. E. Knuth, and O. Patashnik (1994). *Concrete Mathematics: A Foundation for Computer Science.* Reading, Mass: Addison-Wesley.

Greenberg, J., K. M. Spurgin, and A. Crystal (2006). Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. *International Journal of Metadata, Semantics and Ontologies 1*(1), 3–20.

Griffiths, J.-M. and D. W. King (2002). US Information Retrieval System Evolution and Evaluation. *IEEE Annals of the History of Computing 24*(3), 35–55.

Gross, J. L. (2008). *Combinatorial Methods with Computer Applications.* Boca Raton, FL: Chapman & Hall/CRC.

Grossman, D. A. and O. Frieder (2004). *Information Retrieval: Algorithms and Heuristics* (2nd ed.). The Kluwer International Series on Information Retrieval. Springer.

Hafer, M. A. and S. F. Weiss (1974). Word segmentation by letter successor varieties. *Information Storage and Retrieval 10*, 371–385.

Harada, S., M. Naaman, Y. J. Song, Q. Wang, and A. Paepcke (2004). Lost in memories: Interacting with photo collections on PDAs. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, New York, NY, USA, pp. 325–333. ACM Press.

Harary, F. (1959, May). Review: John Riordan, an introduction to combinatorial analysis. *Bulletin of the American Mathematical Society 65*(3), 166–169.

Harbison, S. P. and G. L. Steele (2002). *C: A Reference Manual* (5th ed.). Upper Saddle River, N.J.: Prentice-Hall.

Harman, D. K. (2005). The TREC Ad Hoc Experiments. In E. Voorhees and D. K. Harman (Eds.), *TREC: experiment and evaluation in information retrieval*, Digital libraries and electronic publishing, Chapter 3, pp. 79–97. Cambridge, Mass.: MIT Press.

Harris, J. W. and H. Stöcker (1998). *Handbook of Mathematics and Computational Science*. New York: Springer.

Harter, S. and C. Hert (1997). Evaluation of information retrieval systems: Approaches, issues, and methods. In M. E. Williams (Ed.), *Annual Review of Information Science and Technology (ARIST)*, Volume 32, pp. 3–33. Information Today, Inc.

Harter, S. P. (1975a, July/August). A probabilistic approach to automatic keyword indexing, part I: On the distribution of specialty words in a technical literature. *American Society for Information Science 26*(4), 197–206.

Harter, S. P. (1975b, September/October). A probabilistic approach to automatic keyword indexing, part II: An algorithm for probabilistic indexing. *American Society for Information Science 26*(5), 280–289.

Heine, M. D. (1981). Simulation, and simulation experiments. In K. Sparck Jones (Ed.), *Information Retrieval Experiment*, pp. 179–198. Butterworths.

Hersh, W. R. (2003). *Information Retrieval: A Health and Biomedical Perspective*, Volume Health informatics. New York: Springer.

Hoch, R. (1994). Using IR techniques for text classification in document analysis. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 31–40. Springer-Verlag New York, Inc.

Hozo, S., B. Djulbegovic, and I. Hozo (2005). Estimating the mean and variance from the median, range, and the size of a sample. *BMC Medical Research Methodology 5*(1), 13.

Jacobson, I., G. Booch, and J. Rumbaugh (1999). *The Unified Software Development Process*. The Addison-Wesley object technology series. Reading, Mass: Addison-Wesley.

Jansen, B. J., A. Spink, J. Bateman, and T. Saracevic (1998). Real life information retrieval: A study of user queries on the web. *SIGIR Forum 32*(1), 5–17.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition.* Language, speech, and communication. Cambridge, Mass: MIT Press.

Johnson, B. and L. B. Christensen (2004). *Educational Research: Quantitative, Qualitative, and Mixed approaches.* Boston: Allyn and Bacon.

Johnson, N. L., A. W. Kemp, and S. Kotz (2005). *Univariate Discrete Distributions.* Hoboken, N.J: Wiley.

Jones, G. A. and J. M. Jones (2000). *Information and Coding Theory*, Volume Springer undergraduate mathematics series. London ; New York: Springer.

Jones, K. S. (1981). *Information Retrieval Experiment.* Butterworths.

Jones, K. S. and C. J. van Rijsbergen (1975). Report on the need for and provision of an "ideal" information retrieval test collection. Technical Report British Library Research and Development Report 5266, University of Cambridge.

Jordan, C. (2005). Comparison of blind relevance feedback algorithms using controlled queries. Master's thesis, Dalhousie University Faculty of Computer Science, Canada.

Jordan, C., C. Watters, and Q. Gao (2006). Using controlled query generation to evaluate blind relevance feedback algorithms. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, New York, NY, USA, pp. 286–295. ACM Press.

Kagolovsky, Y. (2003). Terminological problems in information retrieval. *Journal of Medical Systems 27*(5), 399–408.

Kando, N., K. Kuriyama, T. Nozue, K. Eguchi, H. Kato, and S. Hidaka (1999, August). Overview of IR tasks at the first NTCIR workshop. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, Tokyo, japan.

Keen, E. M. (1992). Presenting results of experimental retrieval comparisons. *Information Processing and Management 28*(4), 491–502.

Kekäläinen, J. and K. Järvelin (2002). Using graded relevance assessments in ir evaluation. *Journal of the American Society for Information Science and Technology 53*(13), 1120–1129.

Kluck, M. (2003). Test collection report for the CLEF campaign. Technical Report CLEF-IST-2000-31002, CLEF Language Evaluation Forum.

Knaus, D., E. Mittendorf, P. Schäuble, and P. Sheridan (1995). Highlighting relevant passages for users of the interactive SPIDER retrieval system. In *In TREC-4 Proceedings*.

Knuth, D. E. (1997). *The Art of Computer Programming, Volume 1, Fundamental Algorithms* (3 ed.). Reading, Mass: Addison Wesley Longman.

Knuth, D. E. (2005a). *The Art of Computer Programming, Volume 4, Fascicle 2: Generating All Tuples and Permutations (Art of Computer Programming)*. Addison-Wesley Professional.

Knuth, D. E. (2005b). *The Art of Computer Programming, Volume 4, Fascicle 3: Generating All Combinations and Partitions (Art of Computer Programming)*. Addison-Wesley Professional.

Knuth, D. E. (2006). *The Art of Computer Programming, Volume 4, Fascicle 4: Generating All Trees–History of Combinatorial Generation (Art of Computer Programming)*. Addison-Wesley Professional.

Korfhage, R. R. (1997). *Information Storage and Retrieval*. New York: John Wiley & Sons, Inc.

Kosmala, W. A. J. (1998). *Advanced Calculus: A Friendly Approach*. Upper Saddle River, N.J.: Prentice Hall.

Kraft, D. H. and T. Lee (1979). Stopping rules and their effect on expected search length. *Information Processing and Management 15*(1), 47–58.

Kreher, D. L. and D. R. Stinson (1999). *Combinatorial Algorithms: Generation, Enumeration, and Search*, Volume CRC Press series on discrete mathematics and its applications. Boca Raton, Fla: CRC Press.

Krovetz, R. (1993). Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 191–202. ACM Press.

Lafferty, J. and C. Zhai (2001). Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 111–119. ACM Press.

Lafferty, J. and C. Zhai (2003). *Probabilistic Relevance Models Based on Document and Query Generation*, Volume 13. Kluwer International Series on Information Retrieval.

Lalmas, M. (2005, October). INEX: Evaluating XML retrieval effectiveness. *ERCIM News 63*, 56–56.

Lalmas, M. and A. Tombros (2007). Evaluating XML retrieval effectiveness at INEX. *SIGIR Forum 41*(1), 40–57.

Landi, B., P. Kremer, D. Schibler, and L. Schmitt (1998). Amaryllis: An evaluation experiment on search engines in a french-speaking context. In *Proceeding of the First International Conference on Language Resources & Evaluation LREC. Granada, Spain*, pp. 1211—1214.

Lando, S. K. (2003). *Lectures on Generating Functions.* Number 23 in Student mathematical library. Providence, RI: American Mathematical Society.

Larsen, M. E. (2007). *Summa Summarum.* Ottawa, Ont Wellesley, Mass: Canadian Mathematical Society/Société mathématique du Canada A K Peters.

Lavrenko, V. and W. B. Croft (2001). Relevance-based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 120–127. ACM Press.

Law, A. M. (2006). *Simulation Modeling and Analysis.* Boston: McGraw-Hill.

Lee, C. and G. G. Lee (2005). Probabilistic information retrieval model for a dependency structured indexing system. *Information Processing and Management 41*, 161–175.

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 180–188. ACM Press.

Levy, P. S. and S. Lemeshow (2008). *Sampling of Populations: Methods and Applications*. Hoboken, N.J: Wiley.

Lin, R. T. K., J. L.-T. Chiu, H.-J. Dai, M.-Y. Day, R. T.-H. Tsai, and W.-L. Hsu (2008). Biological question answering with syntactic and semantic feature matching and an improved mean reciprocal ranking measurement. In *IRI*, Proceedings of the IEEE International Conference on Information Reuse and Integration, IRI 2008, 13-15 July 2008, Las Vegas, Nevada, USA, pp. 184–189. IEEE Systems, Man, and Cybernetics Society.

Liu, C. L. (1968). *Introduction to Combinatorial Mathematics*. New York: McGraw-Hill.

Loos, E. E., S. Anderson, D. H. Day, Jr., P. C. Jordan, and J. D. Wingate (2005). *Glossary of Linguistic Terms*. LinguaLinks.

Losee, R. (1987, July). Probabilistic retrieval and coordination level matching. *Journal of the American Society for Information Science 38*(4), 239–244.

Losee, R. M. (1995). Determining information retrieval and filtering performance without experimentation. *Information Processing and Management 31*(4), 555–572.

Losee, R. M. (1998). *Text Retrieval and Filtering: Analytic Models of Performance*. Boston: Kluwer.

Losee, R. M. (2000). When information retrieval measures agree about the relative quality of document rankings. *Journal of the American Society of Information Science 51*(9), 834–840.

Losee, R. M., A. Bookstein, and C. T. Yu (1986). Probabilistic models for document retrieval: A comparison of performance on experimental and synthetic databases. In *SIGIR '86: Proceedings of the 9th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 258–264.

Losee, R. M. and L. Church (2004). Information retrieval with distributed databases:

Analytic models of performance. *IEEE Transactions on Parallel and Distributed Systems 15*(1), 18–27.

Losee, R. M. and L. A. H. Paris (1999). Measuring search engine quality and query difficulty: Ranking with Target and Freestyle. *Journal of the American Society for Information Science 50*(10), 882–889.

Lovász, L. (2007). *Combinatorial Problems and Exercises* (Second ed.). Providence, R.I.: AMS Chelsea Publishing.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics 11*, 22–31.

Luenberger, D. G. (2006). *Information science.* Princeton, NJ: Princeton University Press.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development 1*(4), 309–317.

Manning, C. D., P. Raghavan, and H. Schütze (2008). *An Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Margulis, E. L. (1993). Modelling documents with multiple Poisson distributions. *Information Processing and Management 29*(2), 215–227.

Martin, J. (1990). *Information Engineering Book II: Planning and Analysis (Book 2).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

McFadden, F. and J. Hoffer (1994). *Modern Database Management* (4th ed.). Redwood City, CA.,: Benjamin/Cummings Publishing.

McLachlan, G. J. and D. Peel (2000). *Finite Mixture Models*. New York: Wiley.

McSherry, F. and M. Najork (2008). Computing information retrieval performance measures efficiently in the presence of tied scores. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. W. White (Eds.), *ECIR*, Volume 4956 of *Lecture Notes in Computer Science*, pp. 414–421. Springer.

Meadow, C. T., B. R. Boyce, D. H. Kraft, and C. Barry (2007). *Text Information Retrieval Systems* (Third Edition ed.). Library and Information Science. Burlington, MA: Academic Press.

Mendenhall, W., L. Ott, and R. L. Scheaffer (1971). *Elementary Survey Sampling*. Belmont, CA: Duxberry Press.

Metzler, D. and W. B. Croft (2005). A Markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 472–479. ACM.

Moens, M. (2000). *Automatic Indexing and Abstracting of Document Texts*. Boston: Kluwer Academic Publishers.

Mood, A. M., F. A. Graybill, and D. C. Boes (1973). *Introduction to the Theory of Statistics*. New York: McGraw-Hill.

Moon, S. B. (1993). *Enhancing Performance of Full-Text Retrieval Systems Using Relevance Feedback*. Ph. D. thesis, The University of North Carolina at Chapel Hill, Chapel Hill, N.C.

Mooney, R. J. (2006). CS 371R: Information retrieval and web search. Retrieved April 29, 2006 from `http://www.cs.utexas.edu/~mooney/ir-course/slides/Evaluation.ppt`.

Morecroft, J. D. W. (1988). System dynamics and microworlds for policy makers. *European Journal of Operational Research 35*, 301–320.

Nijenhuis, A. and H. S. Wilf (1978). *Combinatorial Algorithms For Computers and Calculators*. New York: Academic Press.

Olkin, I., L. J. Gleser, and C. Derman (1994). *Probability Models and Applications*. New York, N.Y.: Prentice-Hall College Division.

Paice, C. D. (1990). Another stemmer. *SIGIR Forum 24*(3), 56–61.

Pemmaraju, S. V. and S. S. Skiena (2003). *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Cambridge, U.K. ; New York: Cambridge University Press.

Pepe, M. S. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Number 28 in Oxford statistical science series. Oxford; New York: Oxford University Press.

Ponte, J. M. and W. B. Croft (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 275–281. ACM Press.

Porter, M. F. (1997). An algorithm for suffix stripping. In *Readings in information retrieval*, San Francisco, CA, USA, pp. 313–316. Morgan Kaufmann Publishers Inc.

Pratt, J. W., H. Raiffa, and R. Schlaifer (1995). *Introduction to Statistical Decision Theory*. Cambridge, Mass: MIT Press.

Prikhod'ko, S. M. and E. F. Skorokhod'ko (1982). Automatic abstracting from analysis of links between phrases. *Nauchno-TekhnicheskayaInformatsiya, Seriya2 16*(1), 27–32.

Purdom, P. W. and C. A. Brown (1985). *The Analysis of Algorithms*. New York, N.Y: Holt, Rinehart and Winston.

Raghavan, V. V., P. Bollmann, and G. S. Jung (1989). Retrieval system evaluation using recall and precision: Problems and answers. *SIGIR Forum 23*(SI), 59–68.

Raghavan, V. V., H.-p. Shi, and C. T. Yu (1983). Evaluation of the 2-Poisson model as a basis for using term frequency data in searching. *SIGIR Forum 17*(4), 88–100.

Rakha, M. A. and E. S. El-Sedy (2004). Application of basic hypergeometric series. *Applied Mathematics and Computation 148*, 717–723.

Rasmussen, E. M. (2005). Information retrieval. Retrieved September 12, 2005 from http://www.bookrags.com/sciences/computerscience/information-retrieval-csci-01.html.

Rees, A. M. and D. G. Schultz (1967). A field experimental approach to the study of relevance assessments in relation to document searching. Technical report, Center for Documentation and Communication Research, School of Library Science, Case Western University, Cleveland, OH.

Reingold, E. M., J. Nievergelt, and N. Deo (1977). *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs, N.J: Prentice-Hall.

Riordan, J. (1958). *An Introduction to Combinatorial Analysis*. New York: John Wiley & Sons, Inc.

Rizzo, M. L. (2008). *Statistical Computing with R*. Boca Raton, FL: Chapman & Hall/CRC.

Roberts, F. S. and B. Tesman (2009). *Applied Combinatorics*. Boca Raton, Fla: CRC Press.

Robertson, S. (2001). Evaluation in information retrieval. In M. Agosti, F. Crestani, and G. Pasi (Eds.), *Lecctures in Information Retrieval*, Volume 1980 of *Lecture Notes in Computer Science*, pp. 81–92. New York, NY, USA: Springer-Verlag New York, Inc.

Robertson, S. E. (1974). Specificity and weighted retrieval. *Journal of Documentation 30*, 41–46.

Robertson, S. E. (1981). The methodology of information retrieval experiments. In K. Sparck-Jones (Ed.), *Information retrieval experiment*, pp. 9–31. London: Butterworths.

Robertson, S. E. (1986). On relevance weight estimation and query expansion. *Journal of Documentation 42*(3), 182–188.

Robertson, S. E. (1990). On sample sizes for non-matched-pair IR experiments. *Information Processing and Management 26*(6), 739–753.

Robertson, S. E. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation 60*(5), 503–520.

Robertson, S. E. and K. S. Jones (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science 27*(3), 129–146.

Robertson, S. E. and S. Walker (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 232–241. Springer-Verlag.

Robson, C. (2002). *Real World Research: A Resource for Social Scientists and Practitioner-Researchers.* Oxford, UK ; Madden, Mass: Blackwell Publishers.

Rose, C. and M. D. Smith (2002). *Mathematical Statistics with Mathematica*, Volume Springer texts in statistics. New York: Springer.

Rosen, K. H. (1999). *Discrete Mathematics and Its Applications.* Boston: WCB/McGraw-Hill.

Rosen, K. H. (2005). *Elementary Number Theory and Its Applications.* Boston: Pearson/Addison Wesley.

Rosen, K. H., J. G. Michaels, J. L. Gross, J. W. Grossman, and D. R. Shier (2000). *Handbook of Discrete and Combinatorial Mathematics.* Boca Raton: CRC Press.

Rui, Y., T. S. Huang, M. Ortega, and S. Mehrotra (1999, Fall). Information retrieval beyond the text document. *Library Trends 48*(2), 455–474.

Salton, G. (1975). *A Theory of Indexing.* J. W. Arrowsmith.

Salton, G. (1986). Another look at automatic text-retrieval systems. *Communications of the ACM 29*(7), 648–656.

Salton, G. and C. Buckley (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management 24*(5), 513–523.

Salton, G. and C. Buckley (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science 41*(4), 288–297.

Salton, G. and M. McGill (1983). *Introduction to Modern Information Retrieval.* New York: McGraw-Hill.

Salton, G. and M. Smith (1989). On the application of syntactic methodologies in automatic text analysis. In *SIGIR '89: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 137–150. ACM Press.

Schamber, L. (1994). Relevance and information behavior. In M. E. Williams (Ed.), *Annual Review of Information Science and Technology (ARIST)*, Volume 29, pp. 3–48. Information Today, Inc.

Schamber, L., M. Eisenberg, and M. S. Nilan (1990). A re-examination of relevance: Toward a dynamic, situational definition. *Information Processing and Management 26*(6), 755–776.

Schenck, D. A. and P. R. Wilson (1994). *Information Modeling: The EXPRESS Way*. New York, NY, USA: Oxford University Press, Inc.

Sedgewick, R. and P. Flajolet (1996). *An Introduction to the Analysis of Algorithms*. Reading, Mass: Addison-Wesley.

Shaw, W. M. (1995). Term-relevance computations and perfect retrieval performance. *Information Processing and Management 31*(4), 491–498.

Shaw, W. M., J. B. Wood, R. E. Wood, and H. R. Tibbo (1991). The Cystic Fibrosis Database: Content and research opportunities. *Library and Information Science Research 12*, 347–366.

Siegel, S. (1956). *Nonparametric Statistics for the Social Sciences*. New York: McGraw-Hill.

Slater, L. J. (1966). *Generalized Hypergeometric Functions*. New York: Cambridge University Press.

Solomon, P. (1999). Information mosaics: Patterns of action that structure. In T. Wilson and D. K. Allen (Eds.), *Exploring the contexts of information behaviour*, London, UK, pp. 150–175. Taylor Graham.

Song, I.-Y., M. Evans, and E. K. Park (1995). A comparative analysis of entity-relationship diagrams. *Journal of Computer & Software Engineering 3*(4), 427–459.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application to retrieval. *Journal of Documentation 28*(1), 11—21.

Sparck Jones, K. (2005). Metareflections on TREC. In E. Voorhees and D. K. Harman (Eds.), *TREC: experiment and evaluation in information retrieval*, Digital libraries and electronic publishing, Chapter 3, pp. 421–448. Cambridge, Mass.: MIT Press.

Sparck-Jones, K. and J. R. Galliers (1996). *Evaluating Natural Language Processing Systems: An Analysis and Review*. Berlin ; New York: Springer.

Spärck Jones, K. and C. J. van Rijsbergen (1976). Information retrieval test collections. *Journal of Documentation 32*(1), 59–75.

Spector, P. (2008). *Data Manipulation with R*. New York: Springer.

Spink, A., H. Greisdorf, and J. Bateman (1998, September). From highly relevant to not relevant: Examining different regions of relevance. *Information Processing and Management 34*(5), 599–621.

Spink, A. and R. M. Losee (1996). Feedback in information retrieval. In M. E. Williams (Ed.), *Annual Review of Information Science and Technology (ARIST)*, Volume 31, pp. 33–78. Information Today, Inc.

Srinivasan, P. (1990, January). On generalizing the two-Poisson model. *American Society for Information Science 41*(1), 61–66.

Stanek, W. R. (2002). *XML Pocket Consultant*. Redmond, Wash.: Microsoft Press.

Stanley, R. P. (1997). *Enumerative Combinatorics, Volume 1.* Cambridge studies in advanced mathematics ; 49. Cambridge ; New York: Cambridge University Press.

Sterman, J. D. (1991). A skeptic's guide to computer models. In G. O. Barney (Ed.), *Managing a Nation: The Microcomputer Software Catalog*, pp. 209–229. Boulder, CO: Westview Press.

Stroustrup, B. (2000). *The C++ Programming Language*. Reading, Mass: Addison-Wesley.

Swets, J. A. (1969). Effectiveness of information retrieval methods. *American Documentation 20*(1), 72–89.

Tague, J. (1981). The pragmatics of information retrieval experimentation. In K. S. Jones (Ed.), *Information Retrieval Experiment*, pp. 59–102. Butterworths.

Tague-Sutcliffe, J. (1992). The pragmatics of information retrieval experimentation, revisited. *Information Processing and Management 28*(4), 467–490.

Takaoka, T. (1999). An O(1) time algorithm for generating multiset permutations. In *ISAAC '99: Proceedings of the 10th International Symposium on Algorithms and Computation*, London, UK, pp. 237–246. Springer-Verlag.

Tang, R., W. M. Shaw, and J. L. Vevea (1999). Towards the identification of the optimal number of relevance categories. *Journal of the American Society for Information Science 50*(3), 254–264.

Teorey, T. J. (1991). *Database Modeling and Design: The Entity-Relationship Approach.* San Mateo, CA.: Morgan Kaufmann Kauffmann.

Terrell, G. R. (1999). *Mathematical Statistics: A Unified Introduction*, Volume Springer texts in statistics. New York: Springer.

Trippi, R. R. (1975). Strategies for solving economic problems involving permutations. *Decision Sciences 6*(4), 700–706.

Tucker, A. (1980). *Applied Combinatorics.* New York: Wiley.

Vakkari, P. and N. Hakala (2000, September). Changes in relevance criteria and problem stages in task performance. *Journal of Documentation 56*(5), 540–562.

van Rijsbergen, C. J. (1979). *Information Retrieval.* Newton, MA, USA: Butterworth-Heinemann.

van Rijsbergen, C. J., D. J. Harper, and M. F. Porter (1981). The selection of good search terms. *Information Processing and Management 17*, 77–91.

Velleman, D. J. (1994). *How to Prove It: A Structured Approach.* New York, NY, USA:

Cambridge University Press.

Vogt, C. C. (1999). *Adaptive Combination of Evidence for Information Retrieval*. Ph. D. thesis, University of California, San Diego.

Voorhees, E. (2001). Overview of the question answering track. In *Proceedings of the TREC-10 Conference*, Gaithersburg, MD, pp. 157—165. NIST.

Voorhees, E. and D. K. Harman (2005). *TREC: Experiment and Evaluation in Information Retrieval*. Digital libraries and electronic publishing. Cambridge, Mass.: MIT Press.

Voorhees, E. M. (1999, November). The TREC-8 question answering track report. In E. M. Voorhees and D. K. Harman (Eds.), *Proceedings of the 8th Text REtrieval Conference, Gaithersburg, Maryland, USA*, pp. 77–82.

Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management 36*(5), 697–716.

Voorhees, E. M. (2005, October/November). Trec: Improving information access through evaluation. *Bulletin of the American Society for Information Science and Technology 32*(1), 16–21.

Voorhees, E. M. and D. M. Tice (1999). The TREC-8 question answering track evaluation. In *Proceedings of TREC-8*, pp. 84–106.

Vu, H.-T. and P. Gallinari (2005). On effectiveness measures and relevance functions in ranking inex systems. In G. G. Lee, A. Yamada, H. Meng, and S.-H. Myaeng (Eds.), *AIRS*, Volume 3689 of *Lecture Notes in Computer Science*, pp. 312–327. Springer.

Walpole, R. E. (2002). *Probability and Statistics for Engineers and Scientists*. Upper Saddle River, NJ: Prentice Hall.

Webster's (1996). *Webster's New Universal Unabridged Dictionary*. Barnes and Noble.

Weisstein, E. W. (2003). *CRC Concise Encyclopedia of Mathematics*. Boca Raton: Chapman & Hall/CRC.

White, I. (1994). *Using the Booch Method: A Rational Approach.* Redwood City, CA: Benjamin/Cummings Publishing Company.

Wikipedia (2006). Brown corpus.

Wilbur, W. J. and K. Sirotkin (1992). The automatic identification of stop words. *Journal of Information Science 18*(1), 45–55.

Wilf, H. S. (2006). *Generatingfunctionology.* Wellesley, Mass: A K Peters.

Williams, D. (2001). *Weighing the Odds: A Course in Probability and Statistics.* Cambridge ; New York: Cambridge University Press.

Wolfram, S. (2003). *The Mathematica Book.* Champaign, IL: Wolfram Media.

Yang, Y. and J. Wilbur (1996). Using corpus statistics to remove redundant words in text categorization. *Journal of the American Society for Information Science 47*(5), 357–369.

Yu, C. T., C. Buckley, K. Lam, and G. Salton (1983). A generalized term dependence model in information retrieval. Technical report, Cornell University, Ithaca, NY, USA.

Zhai, C. and J. Lafferty (2001). Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, New York, NY, USA, pp. 403–410. ACM Press.