Alphabet Soup: Learning Knowledge Graphs from Recipes on the Web

Adam Aji, supervised by Tamara Berg

***Abstract*—World knowledge (assertions such as "you can slice an apple") is useful in a variety of applications ranging from annotating video segments to generating motion plans for robots. While people generally already have this kind of knowledge, giving a machine this insight is a challenging problem. However, with the growth of the internet, there has been an increasing amount of data to learn from; people post more pictures and write more text every day. From these, one relatively untapped source of data lies in the domain of cooking, in the form of cooking recipes. As instructional texts, recipes hold information about states of objects and what changes they undergo to reach a goal, and thus present an opportunity to learn world knowledge for applications in automatic illustrations and video annotation. I will present a method to try and extract this information from these data.**

I. INTRODUCTION

With the growth of the internet, there has been an increasing amount of data to learn world knowledge from; people post more pictures and write more text every day. This is a great boon to the major problems in computer vision, like object classification, in that there exists more data for training state-of-the-art models. Specifically, the problem of object classification is this: given a set of images and labels, assign the correct labels to the correct images. Ultimately, this task is also a problem for language and the natural language processing community. Pictures are grounded in and described with words, so the two fields are naturally intertwined.

State of the art methods do fairly well in solving this problem [9]; that is, given the main benchmark, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [19], which uses over a million images across a thousand different labels.

The ILSVRC spans common categories such as cat and dog. However, what if we want to learn about more specific image categories? Or categories that have yet to be seen? These state-of-the-art models only classify between a fixed number of categories (the thousand categories from the ImageNet Large Scale Visual Recognition challenge.) One example of such categories are transition states of objects, like how an "apple" can become a "sliced apple" (example images shown in Figure 1.) Where can we find information about these categories?

From the available sources of data, there exist large databases of objects and their hierarchical and taxonomical relations to categories [16]. While they are expansive in the range of categories that they cover, they are ultimately expensive to create and maintain since they are crafted by hand. There are also objects which fit into categories that we don't know about, so, we would like to look into an alternate source of data for learning.

Figure 1. Two example images of different ingredient states (left: "apple", right: "sliced apple", "slices")

One relatively untapped source of data on the web lies in the domain of cooking. Textual recipes are hosted on a wide range of sites, and the web is full of images of different meals and what goes into them. Furthermore, as instructional texts, recipes hold information about states of objects and what changes they undergo to reach a goal. We can denote these states as newly learned categories.

There are two obvious applications that stem from the connection between learning the knowledge graph, and classifying images of those states. One is the ability to automatically illustrate a given text-only recipe. That is, being able to assign a semantically corresponding image to each step of a recipe. The other application goes in the other direction; that is, given a video (or a set of images) of a recipe being followed, annotate the frames in a way that also matches the composition semantically. Both may be useful in terms of helping people cook, but their real value lies in the insight we gain for instructional texts, which have further applications in fields such as robotics.

I will present a method to try and extract the necessary knowledge graphs from online data, as well as a way to classify images based on them.

II. Background and Related Work

There exist several databases containing knowledge of the world, its structures, and different taxonomies, such as WordNet [16] and ImageNet [8]. WordNet specifically contains entries of words and their definitions, as well as a compilation of each words' hypernyms and hyponyms (i.e. parents and children respectively for *is-a* relationships.) WordNet also includes sets of word synonyms, which are referred to as synsets. ImageNet is a database of over 14M images, which span over several different categories. ImageNet also uses the synsets from WordNet as categories with which to classify their images, and has over 20K synsets indexed.

While these databases are expansive and contain a wide range of valuable information, they are expensive to create, as they are crafted by hand, and also do not contain much information on our categories of interest: transition states.

There have also been attempts to create knowledge bases that learn automatically, like the Never Ending Image Learner [7] and the Never Ending Language Learner [6]. These databases generally learn set relations of objects similar to those found in the earlier WordNet and ImageNet (*is-a* and *has-a* relationships.) Example knowledge might be "cars have wheels", or "apples are fruit". However, there are few (if any) which attempt to portray the different states that an object can take, which is what our work focuses on.

In terms of instructional tasks in general, there exist several studies on generating plans from natural language input [2, 5, 12]. In the space of learning from cooking recipes, one approach uses text only recipes to construct an action graph of the recipe [10]. Their method uses a Bayesian (i.e. probabilistic) approach, and their action graph depicts what actions should be performed on what ingredients in sequence. [17] also looks at a similar style of action graph, but looks specifically to recipes in Japanese. Our method is different from these in that our focus is on the different states of objects, instead of the actions that change them.

There has also been work in looking at cooking videos with accompanying recipe texts [13, 14]. While they achieve the task of auto-illustrating recipes with images, their task is mainly that of alignment; they work on the assumptions that each source of images is directly connected with a recipe, and further, they use the speech in the video for aligning the recipe steps with video frames.

There exist image and video based datasets specifically in the cooking domain, such as [18] and [4]. The first is a collection of videos of people performing basic cooking actions, and the second is a collection of images for different types of dishes. However, both have a different focus from our approach; that is, actions of people and specific dishes versus the states that ingredients go through.

Lastly, understanding of instructional texts also have numerous applications, especially in the field of robotics. In the cooking domain, [3] demonstrates a robot which completes a baking task.

III. Method

The following method can be divided into two main segments, one for dealing with the language in the textual recipes, and the other for the images of ingredient states. Each of those can be broken further into data collection and analysis. I will detail each of those in the following sections.

A. Data Collection (Recipes)

Towards learning about all kinds of recipes and ingredients, we first want to collect text-based recipes. Specifically, we look at recipes which have at least one ingredient from a predetermined list of basic food items, like apple or tomato. We look on recipe.com and query for each ingredient, and scrape at most the top 500 recipes (to keep recipes which are unrelated to the query out of our collection), discounting repeat recipes. For 21 ingredients, this gives us over 6,000 recipes.

At first, we intended to find recipes which also include pictures of the instructions at each step. However, at the time of writing, there just aren't that many cooking sites or blogs which post these kinds of recipes. Despite this, we also attempt to collect these recipes. We collect about 1,000 recipes from two sites (kayotic.nl and visualrecipes.com), with about 14,000 step-by-step images.

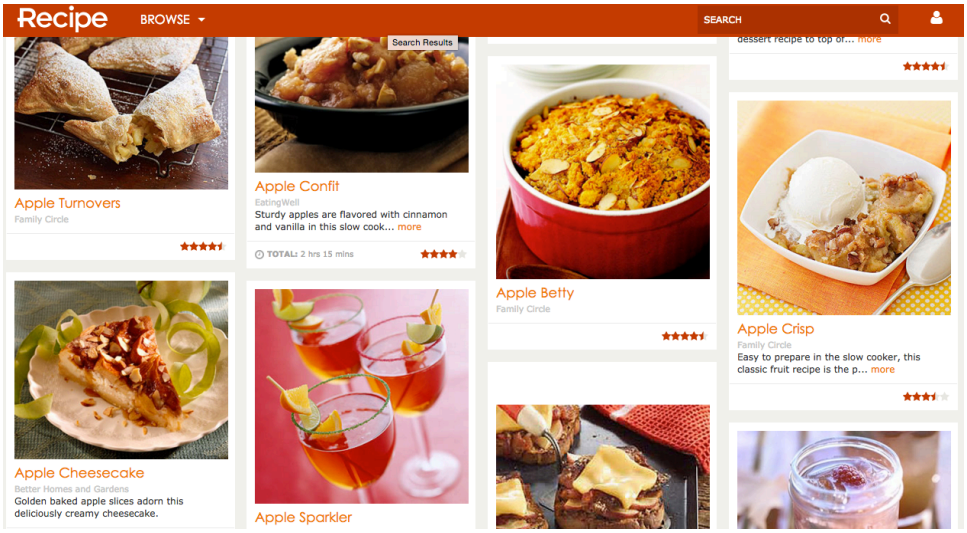

Figure 2. A sample of "apple" recipes from recipe.com

B. Recipe Parsing (Creating knowledge graphs)

In looking at the recipe texts, we would like to extract objects, their different states, and the actions which move these objects through their states. Ultimately, our goal is to create a graph $G = (E, V)$ in which the vertices V denote the states of each ingredient, and the edges E denote the actions used to take objects from one state to another. To this end, we use the Stanford NLP pipeline [15] to do part-of-speech (POS) tagging as well as constituency and dependency parses on the 7K recipes. We first separate each recipe into their corresponding "Ingredients" and "Instructions" sections. We look at each sentence in the "Instructions" section as being a separate step in the instructional process, and run the pipeline on these sentences.

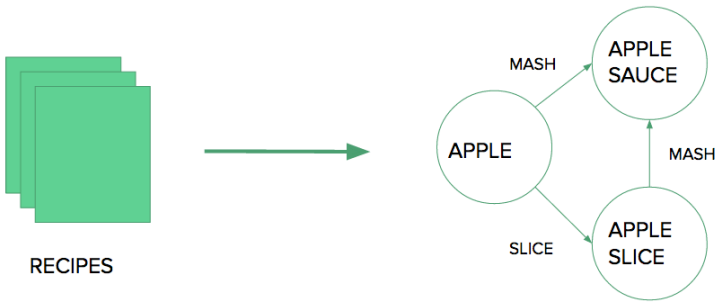

Figure 3. A rough sketch of a knowledge graph for one ingredient.

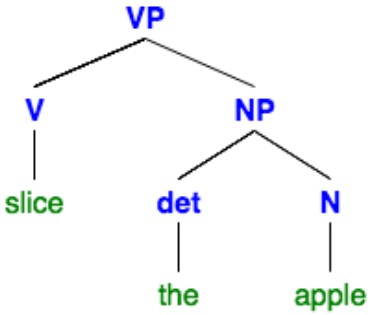

Figure 3. An example syntactic tree for the verb phrase "slice the apple". Constituency parses return groupings such as VP and NP. Visualization generated on < http://mshang.ca/syntree/>.

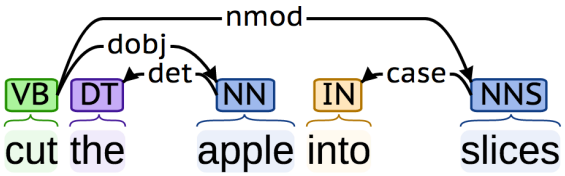

Figure 4. An example dependency parse for the phrase "cut the apple into slices". In this case, we find that "apple" is the direct object of the verb "cut".

POS tagging returns, for each word, a certain POS (e.g. noun, verb, adjective, etc.) Constituency parses give us which words are grouped together in phrases, and which phrases are grouped together to make trees (e.g. “dog ran” as a verb phrase.) The dependency parses give us additional information on certain dependencies between pairs of words, where the most interesting relation to us is being a “direct object”. Out of these NLP tasks, we cannot be certain of their overall accuracy, as the model we use with the Stanford parser is trained on newswire text, which is not very representative of recipe texts.

Ultimately, what we want to do is parse these sentences into “micro” steps, that is, steps which contain the smallest semantic increment in the whole instruction. Each micro-step should contain some action, along with the objects it acts on. From the micro-steps, we can create a graph of actions between object states. To do this, we first look for sentences that don’t have verbs, and toss them out, since we want steps with actions. Then, we look for the verb phrases in the remaining sentences; if we have a sentence with a conjunction (e.g. and), with a verb phrase as its parent, we split that sentence at the conjunction to make two smaller steps, since we want the smallest semantic increments.

From our steps, we create a knowledge graph by following the chain of objects and actions that are acted upon them. First, we know that the states of the objects in the ingredient list must be states (by virtue of starting with them) and that they must be described “decently” to be starting states (for clarity of the recipe.) We then look for these states in the parsed micro-steps, and link each by the corresponding action in the micro-step.

C. Data Collection (Images)

In terms of the visual representation of our model, we also collect images to describe the objects that we are learning about.

To start, we collect images of base ingredients (e.g. “apple”, not “sliced apple”) from ImageNet. We retrieve the labels from the collected 7K set of recipes by looking at nouns and noun phrases. Specifically, for the base ingredients, we look only at the nouns (since bases should not use any modifying attributes.) We then use WordNet to determine if these nouns are “food”, by looking up the hypernym tree of each word and finding the “food” synset. We start with 180 different categories, and across these, we collect over 110K images.

We plan to collect images for all classes of nouns and noun phrases found in the original 7K set, as well as determine which of these classes are not feasible for the task at hand (if the queried result images have little to no correspondence to the actual query.)

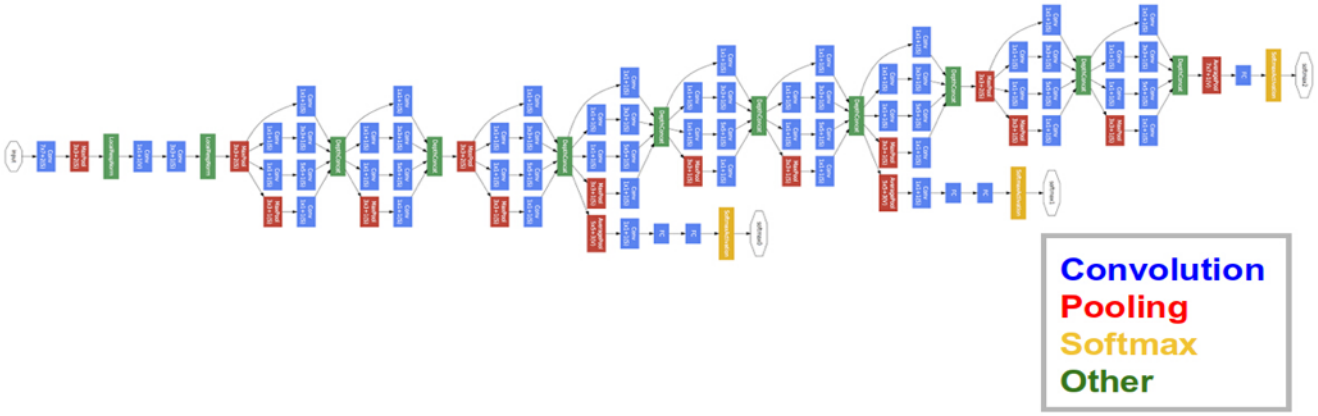

Figure 5. The architecture of the GoogLeNet Inception model. From <http://www.gageet.com/wp-content/uploads/2014/09/googlenet.jpg>

D. Image Classification

From this, we would like to train a visual model to classify based on our labels. There are several methods to do this, but more recently people have had success with deep learning approaches [9]. We can also use models that have been pre-trained for ILSVRC as a starting point. This allows us to reuse some of their already-learned lower level features, and speed up the training. In particular, we look at the GoogLeNet Inception model [20], shown in Figure 6, as it is easier to implement with the deep learning framework we are working with (specifically, TensorFlow [1].)

We fine-tune the Inception model to classify between the categories given by the nouns and noun phrases. Later, we may like to try a hierarchical approach to the classification; first, by classifying between the simple ingredients, then, on each of those subclasses, classifying between states (chopped, sliced, etc.)

From the visual model, we can align the states in the knowledge graph with the classifications, such that we can parse a new recipe text and annotate each micro-step with a series of images.

Figure 7. Examples of misclassified images. Left: most probable "banana", expected “apple”; right: most probable: "celery", expected “apple”

IV. Results and Discussion

In terms of the knowledge graph, we get mixed results. As an example, let’s look specifically at the “apple” class of categories. Figure 1 shows some of the sample states that we find while parsing the “apple” recipes. We find states which we expect, such as “slices” and “chopped”, but we also find weird states that don’t seem to fit the schema, like “apple cinnamon flavored yogurt” and “place the apple in”.

Errors like the first are a result of looking at any and all NPs that are related to the original “apple”, and may be trimmed by incorporating a maximum depth for the syntactic tree of each NP (i.e. by cutting off certain prepositional phrases.) Errors like the second are due to the inaccuracy of the StanfordNLP parser. This is because the models we use with the parser are trained on newswire text (e.g. CoNLL), whereas the evaluated text here are recipes. The text between news and recipes are ultimately very different; in recipes, sentences are often fragments and objects are often left unpronounced.

For the image classification, we refer to preliminary results on training a classifier across 180 categories of “base” ingredients (i.e. ingredients in their whole states.) We find that the Top-1 error (with respect to classifying the image as the most probable class) from evaluation on these images to be 49.3%. While most classification papers present their Top-5 error as the standard metric by which to measure accuracy, we believe that the Top-1 error in this scenario is the most important, since the domain is restricted to a subset of the space of existing “objects”, and a slight change in ingredients can ruin an entire dish (e.g. a cherry looks like a tomato at a different scale, but most certainly does not taste like a tomato.)

In terms of comparison, we actually do not do very well with respect to the other state-of-the-art methods. Other methods achieve around 30-40% Top-1 errors [9]. Figure 7 shows some of the images which we misclassify.

V. Future Work

I plan to extend this work by looking at videos as a source of data, and expanding on the complexity of our language analysis. The methods used will be mostly similar to those previously mentioned, however, using video frames in addition to images. I am currently not working with video due to time constraints, but there are several gains to be made in using them. For example, videos contain images of mixtures of ingredients that are harder to find in online queries.

Furthermore, videos contain images of actions being performed on ingredients, whereas elsewhere on the web, it is harder to find ingredients in these "middle states". As for language, I will look into quantifying phrases (i.e. phrases which specify how much of an ingredient to use), since quantities shown in video should be more precise.

It will also benefit us to take our method outside of the cooking domain; generic "how-to" guides seem to be a valuable place to start, as most "how-to" content is user generated, and spans a wide range of topics (e.g. how to build a table.) Examples of this data can be found on websites like instructables.com.

VI. Conclusion

Natural language understanding is a highly challenging problem, and there is still much left to be done. We show that cooking recipes can be parsed for information of transitional states of ingredients, which can be used for further image classification. We also show some challenges in developing an end-to-end framework for parsing and illustrating recipes.

VII. Acknowledgements

I would like to thank Tamara Berg and Alex Berg for being great mentors and for their guidance in this project. I would also like to thank Miko Marquez for his help and discussions.

References

[1] Abadi, Martın, et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." *arXiv preprint arXiv:1603.04467*(2016).

[2] Artzi, Yoav, and Luke Zettlemoyer. "Weakly supervised learning of semantic parsers for mapping instructions to actions." *Transactions of the Association for Computational Linguistics* 1 (2013): 49-62.

[3] Bollini, Mario, Jennifer Barry, and Daniela Rus. "Bakebot: Baking cookies with the pr2." The PR2 workshop: results, challenges and lessons learned in advancing robots with a common platform, IROS. 2011.

[4] Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool. "Food-101–mining discriminative components with random forests." *Computer Vision–ECCV 2014*. Springer International Publishing, 2014. 446-461.

[5] Branavan, Satchuthananthavale RK, et al. "Reinforcement learning for mapping instructions to actions." Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. Association for Computational Linguistics, 2009.

[6] Carlson, Andrew, et al. "Toward an Architecture for Never-Ending Language Learning." *AAAI*. Vol. 5. 2010.

[7] Chen, Xinlei, Ashish Shrivastava, and Arpan Gupta. "NEIL: Extracting Visual Knowledge from Web Data." *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013.

[8] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[9] He, Kaiming, et al. "Deep Residual Learning for Image Recognition." *arXiv preprint arXiv:1512.03385* (2015).

[10] Kiddon, Chloé, et al. "Mise en Place: Unsupervised Interpretation of Instructional Recipes." *EMNLP 2015*.

[11] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[12] MacMahon, Matt, Brian Stankiewicz, and Benjamin Kuipers. "Walk the talk: Connecting language, knowledge, and action in route instructions." *Def* 2.6 (2006): 4.

[13] Malmaud, Jon, et al. "Cooking with semantics." Proceedings of the ACL 2014 Workshop on Semantic Parsing. 2014.

[14] Malmaud, Jonathan, et al. "What's Cookin'? Interpreting Cooking Videos using Text, Speech and Vision." *arXiv preprint arXiv:1503.01558* (2015).

[15] Manning, Christopher D., et al. "The Stanford CoreNLP Natural Language Processing Toolkit." *ACL (System Demonstrations)*. 2014.

[16] Miller, George A. "WordNet: a lexical database for English." *Communications of the ACM* 38.11 (1995): 39-41.

[17] Mori, Shinsuke, et al. "Flow Graph Corpus from Recipe Texts." *LREC*. 2014.

[18] Rohrbach, Marcus, et al. "Script data for attribute-based recognition of composite activities." *Computer Vision–ECCV 2012*. Springer Berlin Heidelberg, 2012. 144-157.

[19] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision* 115.3 (2015): 211-252.

[20] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.