Cooking Tools Retrival for Recipe Recommendation

|  |
| --- |
|  |
| Qinyu Feng, Bei Su, Kailin Zheng  Computer Science Department  New York University  New York, NY 10010, USA |

Abstract

Recipe recommendation systems typically select the most appropriate recipe for users based on users’ ingredients and cuisine preference. However, the cooking instructions sometimes cannot be followed when people do not have the required cooking tools. In this paper, we propose a cooking tools retrieval method that can extract necessary cooking tools according to the ingredients and instructions in the recipes. The source code of this project can be found on Github via this link:

https://github.com/kz882/recipe

Introduction

Cooking recipes are collaboratively shared knowledge that can be easily searched up online. There are many commercialized APPs and websites available in the market that provides recipes recommendation for users base on ingredients search. For example, Supercook, a recipe search engine, categorizes ingredients into dairies, meat, vegetables, etc. and recommends recipes based on user input of ingredients. Normally a recipe would include ingredients and instructions. But they may not explicitly list the cooking tools necessary for the dishes. However, cooking tools are as important, if not more, as ingredients. Sometimes necessary cooking tools such as aluminum foil and casserole dish may not be at hand upfront.

In this situation, it would be better if the recipe recommendation systems could consider tools that the users have. To help with the recommendation, we develop a Cooking Tools Extraction System that extracts the required tools based on the nouns or noun groups that appear in the instructions. Our first approach is to use Part-Of-Speech tagging to extract the nouns and then eliminate the ingredients from the list of nouns. Our second approach is to compare the rest of the nouns with our knowledge base, a trained dictionary of cooking tools and deem them as results.

There are several challenges in the cooking tools retrieval tasks: first is that the current popular corpora used to train parsers are news articles, which is under different context comparing to cooking recipes. Therefore, the parsers do not work well with our dataset. To resolve this issue, we implement domain adaptation. The second challenge is that it is difficult to identify words that are related to food or not. For example, the word “potato” in “potato masher” is part of the named entity instead of representing a food. We did not implement any method to solve this issue due to time constraint, but we are planning to use named entities in our future work. For evaluating our cooking tools retrieval method, we use precision, recall, and F-measure on our results by comparing the system output with the answers given by our human annotators.

Related Work

Mori et. al. converts the instructions of recipes as procedural texts into directed acyclic graphs as flow graphs to achieve a real “understanding” of the instructions despite the different expressions of human languages. They categorize the words in instructions into named entities (NEs) including food (F), tool (T), duration (D), quantity (Q), action by the chef (Ac), action by food (Af), state of foods (Sf) and state of tools (St). For our purpose, the NEs labeled “food” could be further analyzed and excerpted into ingredients and the actions could be associated with cooking utensils.

Gunamgari et. al. present a more complex tagset for annotating recipes, which is hierarchical and recursive, with higher levels covering broader categories and lower levels covering specific tags. They propose more specific named entities, including Name of the Ingredient (NOI), Properties of Ingredients (POI), Total Size of the Ingredient (TSOI), Total Quantity of the Ingredient (TQOI), preprocessing actions performed on the Ingredients. Cooking steps are further divided into different specifications such as Cooking Action Specifications (ca), Utensils Specifications (uca), Utensils Specifications (uca),  and Time Details (time).

Teng et. al. apply a more straightforward method to extract ingredients. They use regular expression matching to remove non-ingredient terms from the line and identify the remainder as the ingredient. They remove quantifiers, such as “1 lb” or “2 cups”, words referring to consistency or temperature, such as “chopped” or “cold”, along with a few other heuristics, such as removing the content in parentheses. For example “1 (28 ounce) can baked beans (such as Bush’s Original )” is identified as “baked beans”. They then generated an ingredient list sorted by frequency of ingredient occurrence and selected the top 1000 common ingredient names as our finalized ingredient list (Teng). We could use this method to more directly extract the essential information of ingredients to process our noun groups from instructions.

Agarwal and Duong apply Maximum entropy classification of named entities (MaxEnt classifier), semantic role labeling (SRL) and coreference resolution to “identify which actions are applied to which ingredients, and possibly identify which utensils are being used” (Agarwal). Since they only manually labeled 12 recipes for Named Entity Resolution (NER), “it becomes a major source of error for SRL, which in turn would cause errors in coreference resolution” and having a more accurate NER would make the other pieces of their project more accurate. To generate more accurate NER results, our system seeks to improve the Part-Of-Speech tagging by domain adaptation from the Wall Street Journal to the recipes.

Honnibal uses Averaged Perceptron, which adapts the idea of supervised learning to apply a simple learning algorithm to assign different weights to features of each word and average them (Honnibal).

Daumé III proposes to leverage the large, annotated dataset from the source domain and a small, annotated dataset from the target domain. He adds more weight to the dataset of the target domain to achieve feature augmentation for domain adaptation (Daumé III).

Data

## Recipe Box, Structured Recipes Scraped from Food Website

We download our dataset of ~125,000 recipes from Recipe Box, “Structured recipes scraped from food website” as JSON file from eightportions.com (Lee). The description of the dataset says that each recipe consists of: “a recipe title, a list of ingredients and measurements, instructions for preparation, and a picture of the resulting dish”. Since the original dataset was scraped from various food websites, the JSON file contains some unrelated content such as advertisements. We use regular expressions to remove trivial characters.

Example recipe file in JSON:

{'title': 'Slow Cooker Chicken and Dumplings',

  'ingredients': ['4 skinless, boneless chicken breast halves ADVERTISEMENT',

   '2 tablespoons butter ADVERTISEMENT',

   '2 (10.75 ounce) cans condensed cream of chicken soup ADVERTISEMENT',

   '1 onion, finely diced ADVERTISEMENT',

   '2 (10 ounce) packages refrigerated biscuit dough, torn into pieces ADVERTISEMENT',

   'ADVERTISEMENT'],

  'instructions': 'Place the chicken, butter, soup, and onion in a slow cooker, and fill with enough water to cover.\nCover, and cook for 5 to 6 hours on High. About 30 minutes before serving, place the torn biscuit dough in the slow cooker. Cook until the dough is no longer raw in the center.\n',

  'picture\_link': '55lznCYBbs2mT8BTx6BTkLhynGHzM.S'}

We read the JSON file into two dictionaries Python that contain titles as keys, ingredients and instructions as values respectively. We divide our datasets into 50 recipes as the training set, 30 recipes as the development set and 20 recipes as the test set.

Training File (WSJ\_02-21.pos) with Original POS Tags

The training file (WSJ\_02-21.pos) consists of about 950K words. Each line consists of a token, a single blank, and the part-of-speech of that token using the Penn Treebank tag set. Sentence boundaries are marked by an empty line (about 40K sentences). The sentences are retrieved from the Wall Street Journal. Training File (WSJ\_02-21.pos) with Original POS Tags

The training file (WSJ\_02-21.pos) consists of about 950K words. Each line consists of a token, a single blank, and the part-of-speech of that token using the Penn Treebank tag set. Sentence boundaries are marked by an empty line (about 40K sentences). The sentences are retrieved from the Wall Street Journal.

A Non-exhaustive List of Tools as the Knowledge Base

We compile a list of 100 tools retrieved from the three websites: “List of food preparation utensils”, “Kitchen Essentials List: 71 of the best kitchen cookware, utensils, tools & more” and “The Ultimate List of Kitchen Tools for Healthy Cooking”. Each line contains a tool.

Method and Implementation

Part-Of-Speech Tagging

Part-Of-Speech of a token refers to the role it plays in the syntactic function of a sentence. We use the “Alphabetical list of part-of-speech tags used in the Penn Treebank Project” as the definitions of our POS tags. We first apply the nltk.tag.perceptronpackage by Honnibal and Duong to the 50 recipes in the training set as the first step of the POS tags. Based on the POS tags given by the program, the annotators (two of our teammates) manually correct them. Since most of the sentences in the instructions are imperative, the nltk package tends to label the verbs as nouns, for example, in the instruction “Stir in the flour and salt, and simmer until bubbly,” “Stir” is incorrectly labeled as a noun and “simmer” is labeled as a verb. By correcting the noun/verb labels and other labels, we get more accurate POS tags and thus have a more accurate list of nouns for future processing.

Domain Adaptation by Feature Augmentation

We first use the Wall Street Journal corpora with POS tags as the training data. One issue with this dataset is that it contains mostly declarative sentences with syntactic and POS tags that are less reliable in the context of re

cipe instructions. In order to resolve this issue, we apply domain adaptation by adding manually corrected instructions in the training set of 50 recipes to the Wall Street Journal corpora. Since the size of the recipe dataset is not comparable to the WSJ corpora, we perform feature augmentation in the preprocessing step by adding the recipe data 15 times to make it equally weighted as the Wall Street Journal corpora.

Viterbi Algorithm (Hidden Markov Model)

The Viterbi algorithm is a [dynamic programming](https://en.wikipedia.org/wiki/Dynamic_programming) [algorithm](https://en.wikipedia.org/wiki/Algorithm) for finding the most [likely](https://en.wikipedia.org/wiki/Likelihood_function) sequence of hidden states—called the Viterbi path—that results in a sequence of observed events, especially in the context of [Markov information sources](https://en.wikipedia.org/wiki/Markov_information_source) and [hidden Markov models](https://en.wikipedia.org/wiki/Hidden_Markov_model) (HMM) to generate POS tags based on training set (“Viterbi”). We label the development set by applying the Viterbi algorithm on the training set.

Comparing the List of Nouns with Knowledge of Ingredients and Tools

First, extract a list of ingredients stripped off of quantifiers, state of ingredients, etc from the ingredients list of our training set. For example, based off of the ingredients list of the sample JSON file, we expect to generate “chicken breast”, “butter”, “condensed cream”, “onion”, “biscuit dough”. Then, remove the ingredients from the list of nouns. The ingredient section of the recipe contains the list of food and quantity needed for the dish. Since food is included in the noun list that we generated, we remove them from the list to eliminate the incorrect answers in the list. Second, we compare the rest of the nouns with our knowledge base, a trained dictionary of cooking tools and deem as results.

Evaluation

From previous steps, our system generates a list of tools based on the noun groups in the instructions section of the recipes, known as "system output". The human annotators agree on a list of tools known as "answer key". For the two lists, we build a python program that automatically reads the inputs and compares the results, based on the 20 recipes in the test set. For each recipe, we have a precision, recall, f-1 score. Precision is defined as the number of correct tools/the number of answer keys. Recall is the number of correct tools/number of system output. F-measure = 2/ (1/precision + 1/recall).

Future Work

Due to time constraint, several other approaches that can potentially optimize our algorithm are not implemented in the current version of our system.

With time permitting, we would like to generate a more exhaustive list of ingredients and tools for a better knowledge base. Our current system strips the ingredients completely off of the adjectives. However, to prepare the ingredients, some tools may be necessary to pre-process the ingredients before we move on to the instructions section. For example, in the sample JSON recipe file, we may need boning knife and peeling knife to strip chicken breast off of bones and skin. We may also need a special knife to “finely dice” the onion, and use the fridge to freeze the biscuit dough, etc.

Now we retrieve ingredients based on the nouns in the instructions. An updated system speculates the tools based on the action verbs. We first pre-train the list of tools (our current knowledge base) to associate each tool with the action verbs. To do this, we can scrape the description of the tools from their respective Wikipedia page or definitions in the Oxford English. We will calculate the percentage of words that occur together in the same file, or that occur in the same sentence. Then we will associate the tool with the top few action verbs in its description and obtain the Pairwise Mutual Information. Alternatively, we can use the Google BERT system to do this task with machine learning via word-embedding.

References

Alphabetical list of part-of-speech tags used in the Penn Treebank Project:

<https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html>

Agarwal, Rahul. Miller, Kevin. “Information Extraction from Recipes.”

<https://nlp.stanford.edu/courses/cs224n/2011/reports/rahul1-kjmiller.pdf>

Chun-Yuen Teng, Yu-Ru Lin, and Lada A. Adamic. 2012. Recipe Recommendation Using Ingredient Networks. In Proceedings of the 3rd Annual ACM Web Science Conference, ACM, pages. 298-307.

Daumé III, Hal. “Frustratingly Easy Domain Adaptation”. 2009. <https://arxiv.org/pdf/0907.1815.pdf>

Gunamgari, Sharath Reddy, Sandipan Dandapat, and Monojit Choudhury. "Hierarchical Recursive Tagset for Annotating Cooking Recipes." In Proceedings of the 11th International Conference on Natural Language Processing, pp. 353-361. 2014.

Honnibal, Matthew. Duong, Long. Source code for nltk.tag.perceptron.

<https://www.nltk.org/_modules/nltk/tag/perceptron.html>

Honnibal, Matthew. A Good Part-of-Speech Tagger in about 200 Lines of Python. <https://explosion.ai/blog/part-of-speech-pos-tagger-in-python>

Kitchen Essentials List: 71 of the best kitchen cookware, utensils, tools & more. <https://www.mealime.com/kitchen-essentials-list>

Lee, Ryan T. Personal Data Science blog “Eight Portions”project. <https://eightportions.com/datasets/Recipes/#fn:1>

N. Shino, R. Yamanishi and J. Fukumoto, "Recommendation System for Alternative-Ingredients Based on Co-occurrence Relation on Recipe Database and the Ingredient Category," 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Kumamoto, 2016, pp. 173-178.

Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, Tetsuro Sasada. “Flow Graph Corpus from Recipe Texts”. [Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)](https://www.aclweb.org/anthology/volumes/L14-1/). 2014. European Language Resources Association (ELRA) Pp. 2370–2377.

SuperCook. https://www.supercook.com/#/recipes

Ueda, Mayumi & Asanuma, Syungo & Miyawaki, Yusuke & Nakajima, Shinsuke. (2014). Recipe Recommendation Method by Considering the User's Preference and Ingredient Quantity of Target Recipe. Lecture Notes in Engineering and Computer Science. 2209. 519-523.

The Ultimate List of Kitchen Tools for Healthy Cooking.https://greatist.com/eat/ultimate-list-kitchen-tools-healthy-cooking#1

“Viterbi Algorithm.” *Wikipedia*, Wikimedia Foundation, en.wikipedia.org/wiki/Viterbi\_algorithm.

Wikipedia page. List of food preparation utensils.https://en.wikipedia.org/wiki/List\_of\_food\_preparation\_utensils