W266_Project_Milestone_Report

July 26, 2017

Roy Gvirtsman roy.gvirtsman@berkeley.edu
Chuqing He chqngh@berkeley.edu
Tony Panza apanza@berkeley.edu

1 Abstract

This shall examine vector representations of cooking recipes.

2 Introduction

3 Background

A similar work and idea for inspiration is the Stanford CS224n paper by Agarwal and Miller (2011).

4 Methods

4.1 Data Acquisition

Scrapy was used to crawl and scrape all of the recipe data from allrecipes.com into JSON format. To keep the JSON file sizes reasonably small, the scraping was divided into 1 or 2 categories at a time. All of the JSON files were then uploaded into an Amazon AWS S3 bucket. So far, we have accumulated 174,225,333 bytes of recipe data in JSON format.

4.2 Data Processing and Ingredient Extraction - Brute Force

For initial development purposes, only one JSON file (consisting of two categories from allrecipes.com) was loaded into a Pandas dataframe. (Ultimately, all of the JSON files will need to be loaded into a common data frame.)

An example of the `ingredients` column from one of the rows looks like this:

```
In [3]: [u'3/4 cup sweetened dried cranberries, chopped',
         u'1 McIntosh apple - peeled, cored, and diced',
         u'1/2 small red onion, finely chopped',
         u'2 tablespoons lemon juice',
         u'2 teaspoons honey',
         u'1 teaspoon chili powder',
         u'1/2 teaspoon ground cinnamon',
         u'1 (6 ounce) bag baby spinach, torn into bite-sized pieces',
         u'Add all ingredients to list',
         u'Add all ingredients to list']
```

```
Out[3]: [u'3/4 cup sweetened dried cranberries, chopped',
         u'1 McIntosh apple - peeled, cored, and diced',
         u'1/2 small red onion, finely chopped',
         u'2 tablespoons lemon juice',
         u'2 teaspoons honey',
         u'1 teaspoon chili powder',
         u'1/2 teaspoon ground cinnamon',
         u'1 (6 ounce) bag baby spinach, torn into bite-sized pieces',
         u'Add all ingredients to list',
         u'Add all ingredients to list']
```

4.3 Building Ingredient Vocabulary

Each row of the dataframe was run through a function to extract the core ingredients by removing the measurement numbers, units, and descriptions. This was (for now) just done through the use of lookup table to remove unwanted words. The unwanted words are organized into three categories: measurement units, preparatory descriptions, and miscellaneous. Example measurement units are: `cups`, `pounds`, `liters`, `boxes`, and `halves`. Example preparatory descriptions are: `crumbled`, `peeled`, and `thawed`. Example miscellaneous words are: `about`, `thinly`, and `more`. Non-letter characters were also removed.

The example ingredient list shown above was "cleaned" to this:

```
In [4]: [u'sweetened cranberries',
         u'mcintosh apple',
         u'red onion',
         u'lemon juice',
         u'honey',
         u'chili powder',
         u'cinnamon',
         u'baby spinach']
```

```
Out[4]: [u'sweetened cranberries',
         u'mcintosh apple',
         u'red onion',
         u'lemon juice',
         u'honey',
         u'chili powder',
         u'cinnamon',
         u'baby spinach']
```

To get a list of all the unique ingredients from the entire data frame, the cleaned lists from each recipe were flattened into one big list with `np.hstack`, then unique items identified with `set()` function. From this, we have our "vocabulary" of ingredients.

4.4 Vectorized Recipes

Sklearn's CountVectorizer was used to construct a sparse matrix of recipes down the rows and ingredients along the columns. The unique vocabulary for ingredients processing is passed in as the vocabulary argument.

Then sklearn's TruncatedSVD was used to reduce the dimensionality of the recipe-ingredient matrix.

4.5 KMeans Clustering

We also did a simple ingredients clustering on our two JSON categories, appetizers_salads and bbq_bread. The preprocessing step is similar to building the vocabulary step. The KMeans pipeline consist of CountVectorizer, TFIDFTransformer, and KMeans Classifier. We used the categories as the true labels and clustered on 2 centroids. The metrics report is as following:

```
In [7]: #
        '''
                     Precision     Recall   F1-Score
           bbqbread      0.57          0.92      0.70

             salads      0.97          0.78      0.86


        avg / total      0.87          0.81      0.82
        '''
        #

Out[7]: '\n            Precision     Recall   F1-Score     \n    bbqbread     0.57          0.92
```

We also printed the top Words in each cluster:

Top terms per cluster:
Cluster 0 (bbqbread):
pepper, cheese, garlic, salt, black, oil, onion, red, sauce, green

Cluster 1(salads):
flour, baking, sugar, allpurpose, white, milk, butter, powder, salt, eggs,

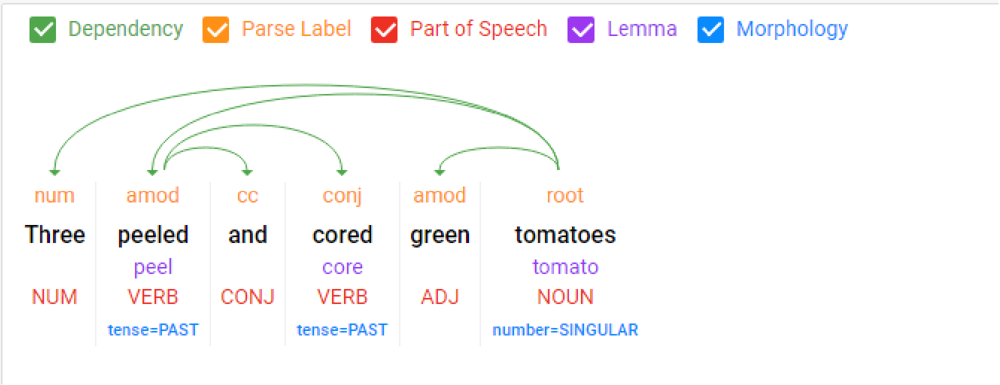
5 Results and Discussion

6 Next Steps

Try the Google part of speech parser as a way to extract and identify the "core" ingredients from an ingredients list.

This would hopefully generalize better than using a lookup table to remove words known to not be of interest.

We also need to do some experimentation and analysis of the optimal n_components used for SVD for dimensionality reduction.



☑ Dependency ☑ Parse Label ☑ Part of Speech ☑ Lemma ☑ Morphology

num	amod	cc	conj	amod	root
Three	peeled	and	cored	green	tomatoes
	peel		core		tomato
NUM	VERB	CONJ	VERB	ADJ	NOUN
	tense=PAST		tense=PAST		number=SINGULAR

Example tagging of ingredients

7 References

Jaan Altosaar. 2017. food2vec - Augmented cooking with machine intelligence – Jaan Altosaar.

Rahul Agarwal and Kevin Miller. 2011. Information Extraction from Recipes. Technical report.

Tiago Simas, Michal Ficek, Albert Diaz-Guilera, Pere Obrador and Pablo R. Rodriguez. 2017. Food-Bridging: A New Network Construction to Unveil the Principles of Cooking. Frontiers in ICT, 4.

Wesley Tansey, Edward Lowe and James Scott. 2016. *Diet2Vec: Multi-scale analysis of massive dietary data*. 1st edition.

Yong-Yeol Ahn, Sebastian E. Ahnert, James P. Bagrow and Albert-László Barabási. 2011. Flavor network and the principles of food pairing. *Scientific Reports*, 1(1).