Dish Recommendations from Yelp Reviews

Anjali Muralidhar, Mike Puncel, Nischay Kumar

Introduction

Yelp.com is a website that allows users to rate and review many local businesses. One of the top segments of these businesses reviewed on Yelp is restaurants. In order to investigate the quality of a particular restaurant, a user can go to Yelp and judge the restaurant based on the number of stars (out of 5) other users have attributed to it in the past. Once users determine if a restaurant is worth visiting, however, they may also wish to determine which dishes on the menu are recommended, in addition to which dishes they should stay away from, so that they can have the best possible dining experience. Currently, the user must scan through several reviews in order to get a sense of which dishes reviewers enjoyed and recommended at a restaurant. The user may have to scroll substantially down the page to read enough reviews that recommend or criticize a dish, and mentally aggregate and average these reviews to figure out which dish is worth trying.

Using natural language processing techniques, the work of aggregating opinions in reviews about a certain dish can be done computationally. The average sentiment about a dish can be evaluated after identifying dishes in reviews and gauging if the dish is described positively or negatively. For a given restaurant, the top dishes can be identified and suggested to the user, without requiring the user to read through the reviews themselves looking for recommendations. Such a feature will allow users of Yelp to not only quickly assess the quality of a restaurant, but also compare the quality of each dish served there.

Restaurant Menus

In order to identify the array of dishes served at a restaurant, we must extract menus from all restaurants that are reviewed in our training and test datasets. Not only do we have to extract the name of all dishes, but we also have to strip descriptive words from the dish names. For example, we abbreviate “homemade gnocchi” to simply “gnocchi” because descriptive words such as “homemade” would likely be omitted in a user review. By reducing dish names to the most concise string without loss of specificity, we are able to better detect mentions of the dish in user reviews.

We also constrain the restaurants used in our dataset to Italian restaurants because the dishes tend to be very uniquely named, also making it easier to identify mentions of a dish in reviews.

Dish Snippet Extraction

As a first step toward ranking dishes at a particular restaurant, we needed to isolate mentions of a particular dish in our dataset. In 6.864 lecture, we learned that an adjective in a sentence is usually within five words of the noun it modifies. The length of an average English sentence is also about 15-20 words. From analysis of user reviews, we also found that the description of the dish usually comes after mentioning the name of the dish. Thus, we decided to look at snippets of the review that consisted of the dish name and the 12 words that followed. This would make it very likely that an adjective that will indicate the quality of the dish will be included

in the snippet, and there will be enough context after the dish name to determine if it is being described positively or negatively.

Sentiment Analysis Using Perceptron

After extracting snippets that describe dishes from restaurant reviews, we needed to determine if the sentiment of the snippet was positive or negative using the perceptron algorithm. In the training set, we label each snippet as positive or negative. Snippets are determined to be positive if the user thought the dish was average or above, while snippets were defined as negative if the user criticized the dish. The perceptron algorithm gives weights to all words in the snippets, and these weights are used to determine the sentiment about snippets in the test dataset. The sentiment about a dish would be aggregated and averaged to determine a general opinion about the dish based on the reviews.

Early Experimental Results

Early experimentation involved training on reviews from 7 restaurants and testing on reviews from 3 restaurants. Manual inspection of the results revealed disappointing performance. Words that were not necessarily a clear indicator of sentiment were given peculiarly high or low weights. For example, the word “you” was given a very high positive weight, although it usually does not indicate a positive opinion.

In order to address these issues, we likely need to train on more restaurants to better assign weights to words. Also, we propose omitting extremely common words like “the,” “and,” “you,” etc. that do not contribute to sentiment. The early experiments also included food words in the vocabulary so a word like “mushroom” was weighted heavily if a certain restaurant had many positive reviews about a mushroom dish. These food words should be omitted from the perceptron algorithm to get better results.

Another technique that may give better results is to tag each word in the reviews with its part of speech and give more weight to adjectives, which are usually the best indication if a dish was described positively or negatively. This would likely lead to the words with the highest weights being words like “tasty” and “delicious” and words with the lowest weights would be adjectives such as “salty” and “undercooked.” There are existing part-of-speech taggers that have been built and vetted by the NLP community. We plan on using one that has shown to tag sentences with accuracy. The tagger built by the Stanford Natural Language Processing Group and discussed in the paper by Toutanova et al. is publicly available for download and is very simple to use. This part-of-speech tagging will allow to us be more refined in defining what words influence the sentiment of a snippet. Empirically adjusting the weights of the tags in our vocabulary will allow us to compare the accuracy of our predictions based on what the weight allocation is for each tag and define a ranking system as well [1].

Currently we are manually going through the set of snippets for each restaurant and labelling a given snippet as positive or negative based on our interpretation of the reviewer’s sentiment. In the paper by Tackstrom and McDonald, an objective approach to labelling sentiments is outlined using the vote-flip algorithm [2]. There are polarity lexicon databases available online including SentiWordNet (http://sentiwordnet.isti.cnr.it/) and MQPA Subjectivity Lexicon (http://www.cs.pitt.edu/mpqa/subj_sense_annotations.html) that define the sentiment associated with a word from the scale of (-1.0 to 1.0). Using these pre-defined sentiments, one can

define thresholds for negative and positive sentiment. We can iterate over the words in a given snippet and count the number of negative and positive words as well as the number of negating modifiers such as “not”. Based on the two sentiment metrics, the vote-flip algorithm determines the consensus sentiment on majority and flips it if the number of negating modifiers is odd. These are preliminary steps that we think will allow us to increase the accuracy and robustness of the perceptron algorithm. However, if we decide this approach isn’t accurate enough for our purposes and time permits, we may look into implementing the conditional random field model using by Tackstrom and McDonald [2].

The results also are much more likely to classify a dish with a positive sentiment rather than a negative one. We suspect that this is because user reviews are more likely to put a dish in a positive light than a negative one. This also suggests that we need more training data. The set we have consists of few negative words, because the proportion of negative snippets to positive ones is much lower than 50%.

Testing and Validation of Results

Yelp offers its users the ability to review the restaurant as a whole, but it does not offer the granular functionality to rate individual dishes at these restaurants. As a result, it was difficult to define a baseline or existing standard to use in comparison with our predictions. We have derived two possible baselines that we could use to determine the performance of our model. The gold standard would be to manually label all of the snippets used in the test data as well and then compare our predictions against those. We can compute statistical measures like sensitivity (True Positives/ (True Positives + False Negatives)) and specificity (True Negatives/ (True Negatives + False Positives)) to determine its performance. We could also perform interesting analysis like clustering snippets of False Negatives and False Positives in order to understand what were the weaknesses of this methodology and possibly suggest improvements for the future.

An alternative baseline to remove subjectivity from our labelling would be to assume all reviews from restaurants with an average rating greater than or equal to 3 stars are positive reviews and all reviews with less than 3 stars are negative reviews. Consequently, since the reviews have been assigned positive and negative labelings by this mapping we can assume that all dishes included in those reviews can be assigned the same label as the review. We could compare the sensitivity and specificity across these two models in reference to the gold standard. We can also change the size of our testing set and determine the precision (True Positives/ (True Positives + False Positives)) and recall (True Positives/ (True Positives + False Negatives)) at each set size, which will allow us to build a precision versus recall curve for our two models. The curves will allow us to compare the robustness of the two models.

Sources

[1] Toutanova et al. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency network. *Proceedings of HLT-NAACL 2003*, pp 252-259.

[2] Tackstrom O. and McDonald R. 2011. Discovering Fine-Grained Sentiment with latent Variable Structured Prediction Models. *SICS technical Report 2011.*