**Abstract**

Users rate and review restaurants on Yelp.com, which then aggregates all user ratings and shows an average rating for that restaurant. This average rating is useful to users who are investigating restaurants and trying to decide where they would like to go. However, once users have isolated a restaurant, it is difficult to assess which dishes are best to order at that restaurant. This paper describes a sentiment analysis system that inputs Yelp restaurant reviews and outputs which dishes are most recommended at a restaurant as well as which dishes, if any, are particularly bad.

**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 dishes served there.

**System Setup**

*Menu Extraction*

In order to identify the array of dishes served at a restaurant, we extracted menus from all restaurants that are reviewed in the training and test datasets. Not only did we have to extract the name of all dishes, but we also had to strip descriptive words from the dish names. For example, we abbreviated “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 constrained 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. However, the sentiment analysis system described in this paper can be extended to any type of restaurant as long as the menu items are properly extracted.

We also required that the restaurants used in our datasets had more than 50 reviews. This was meant to ensure that the restaurant’s reviews would include enough mentions of menu items to train our classifier, since only a fraction of reviews specifically mention the dishes the user ordered.

*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 next 8, 12, or 20 words that followed. This made it very likely that an adjective that indicates the quality of the dish is included in the snippet, and there was enough context after the dish name to determine if it was being described positively or negatively.

The snippets with 8, 12, or 20 words trailing the mention of the food dish are known in the system as short, medium, and long snippets. We introduced this variability in snippet length because, although the adjective to describe a noun appears most often within 5 words of the noun, user reviews usually mentioned the ingredients or garnishing of the dish before declaring a judgment. Thus, we hypothesized that snippets with more trailing words would give a better indication of the user’s sentiment about a dish than snippets with fewer trailing words even though in normal sentences, the descriptive words appear closer to the noun being described.

*Part of Speech Tagging*

We hypothesized that the words that describe a user’s opinion about a dish were most likely the adjectives that he or she used in the review snippet. Thus, we wanted to weight adjectives more heavily when encountered by the sentiment analysis algorithm. We predicted that 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 used 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. [1] is publicly available for download and is very simple to use. This part-of-speech tagging allowed us to be more refined in defining what words influence the sentiment of a snippet. Empirically adjusting the weights of the tagged adjectives in our vocabulary allowed us to compare the accuracy of our predictions between tagged reviews and untagged reviews.

*Sentiment Training*

We utilized supervised methods in order to train our classifier. For each review snippet (short, medium, and long) in our training dataset, we classified the snippet as 1 if the user considered the dish to be average or above average and -1 if the user disliked the dish. These labels were referenced when training the classifier.

We used the same technique to label the snippets derived from the test dataset. These positive or negative labels were used as a “gold standard” in later evaluation of our model.

**Sentiment Analysis**

After extracting snippets that described 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 labeled each snippet as positive or negative. Snippets were determined to be positive if the user thought the dish was average or above average, while snippets were defined as negative if the user criticized the dish. The perceptron algorithm gave weights to words in the snippets, and these weights were used to determine the sentiment about the snippets in the test dataset. The sentiment about a dish was then aggregated and averaged to determine a general opinion about the dish based on the reviews.

We split our overall dataset so that 75% of the data was used as training and 25% as test. There was a total of 31 restaurants evaluated. Twenty-five of these restaurants were used as the training dataset and six restaurants were used as the test dataset, leading to an 80/20 split between training and test data. The 80/20 data split between restaurants corresponded to a 75/25 split between data because the number of reviews per restaurant varied widely and certain restaurants provided a greater number of snippets on which to train the classifier than other restaurants.

We chose to exclude any words that appeared in menu items from the perceptron algorithm. This is because a highly rated mushroom dish would lead to the word “mushroom” being given a high weight, when it is actually the words describing the mushroom dish that we want the perceptron algorithm to weight highly.

When snippets with part of speech tagging were run through the algorithm, words that were tagged as adjectives were given 0.25% more weight than non-adjectives. This was done in order to emphasize adjectives as descriptive words that give the most indication about the sentiment of a review.

It was also interesting to note that there were substantially more positive reviews for dishes than negative ones in our dataset because user are more likely to review a dish in a positive light than a negative one. Thus, even in a relatively large dataset, there are much fewer negative examples of reviews to train on which affected the quality of negative review detection of our algorithm.

**Results and Analysis**

Our version of the perceptron algorithm analyzed user review data in the form of short, medium, and long snippets, as well as tagged and untagged snippets. All results were compared with a gold standard labeling of the test data that was done by humans. They were also compared with a baseline standard labeling of the test data that was automatically generated based on the star rating the user gave to the review.

*Gold Standard*

We manually labeled each snippet in test dataset with a label of 1 if the snippet was neutral to positive, and a label of -1 if the snippet was negative. This labeling was considered a gold standard of how an actual human user would interpret the review snippets. All label predictions generated by the baseline standard, short, medium, and long snippets were compared against the gold standard labels.

*Baseline Standard*

As an automated baseline standard, we labeled snippets in the test dataset based on the star rating given to the review. When each user writes a review for a restaurant, they must also give an overall star rating between 1 and 5 stars. We assumed that a user who enjoyed their dishes at a restaurant would give an overall star rating of 3 or above, while a user who did not enjoy their dishes would give a rating of less than 3 stars. Thus, we give a label of 1 to all snippets that come from reviews that have a star rating of 3 or more, and we give a label of -1 to all snippets that come from reviews with a rating of less than 3.

*Untagged Snippets*

When analyzing review snippets that did not undergo part of speech tagging, all non-food words in the snippets were given equal treatment our perceptron algorithm. Snippets were analyzed in three sizes: small, medium, and long. These snippets were made up of 8, 12, and 20 words after the mention of the food dish, respectively. The short and medium snippets performed nearly as well or worse than the baseline standard when comparing accuracy, precision, recall, specificity, and precision of negative reviews.