Natural Language Processing Instructor: Prof. Ralph Grishman
Term Project Report Student: Ching-Hsiang Hsu, Li-Hao Liu
Deadline: May 10^{th}, 12:05PM

Restaurant Rating System - Evaluating Restaurants by Extracting and Categorizing Information from Reviews

Motivation

Yelp is a very popular application to search for good restaurants. It has 135 million monthly visitors and 95 million reviews in 2016. However, some of the reviews are tediously lengthy and inaccurate, so the information might not live up to your expectation or not match your preference. Therefore, we want to use Natural Language Processing techniques to classify reviews into several categories such as taste, service, price, sanitation, and location. This way, people can have a better understanding from different aspects of the restaurant. In addition, by applying our useful system, these reviews can provide much more objective and fair viewpoints for restaurant runners. For example, if a review has a food category tag and it shows delicious, the owner of the restaurant can refer to this conclusion that his cooks have done a great job. On the other hand, if a review belongs to the environment category and it has a dirty label, the owner of the restaurant needs to ask his waiters and waitresses to be more diligent on maintaining the sanitation.

Moreover, although Yelp provides a rating system for the customers to evaluate the restaurant, the rating system is too general and difficult to learn the detail of the restaurant. Categorizing these reviews can help restaurant owners to have a more comprehensive understanding about their reviews and the quality of the restaurant. For instance, a 3-star restaurant may have 5-star food and 1-star service. We can process these Yelp reviews and extract more information from the reviews. Hence, the restaurant runner can recognize what the problems are and make an improvement base on these precious comments or they can know what their advantages are and put emphasis on their excellent part. One remark is that our restaurant rating system is not Yelp-driven. It can be applied to any kind of website which provides restaurant reviews.

Problem Description

The first problem is what's the objective of this rating system. After several discussion and reading a lot of Yelp's reviews, we define seven basic categories to do annotation and classification which are

1. Overall: LOVE/HATE,
2. Taste: YUMMY/GROSS,
3. Service: FRIENDLY/INHOSPITABLE,
4. Price: CHEAP/EXPENSIVE,
5. Sanitation: CLEAN/DIRTY,
6. Location: CONVENIENT/INCONVENIENT, and
7. Other: OTHER.

The first six categories have positive and negative labels. The positive example is "I like this place very much". "like" will be labeled as LOVE. The negative example is "the restaurant is overrated". The label of "overrated" is HATE. If the token is a natural word, it would be labeled as OTHER. For example, in the sentence, "the food is not good", the word "good" will be labeled OTHER, not LOVE. Here is another selected labeled instance: "Looking for a low budget dining option with delicious food? Burger&Lobster is the choice for you.". "low budget" will be annotated as CHEAP CHEAP and "delicious" will be annotated as YUMMY. Other irrelevant words in the review will be annotated as OTHER.

The second problem is how do we define a reasonable labeling regulation to prevent inconsistent annotation, especially for two or more people who are going to do the annotation tasks. If the labeled data is not consistent, it will influence the result dramatically. We have encountered several annotation ambiguity and conflict. Suppose there are two sentences: "the food is good" and "the food is delicious". After the discussion, the label of "the food is good" is OTHER YUMMY OTHER YUMMY because good describes the food. Therefore, the food has good taste expression. And the label of "the food is delicious" is OTHER OTHER OTHER YUMMY since delicious itself has good taste meaning.

The third problem is once we have these labeled data, what kinds of features we need to add for training to do classification? Combining with what we have learned and experienced in the assignments, we have tried part-of-speech (POS), sequence POS(sequence means trigram or more tuples), POS combination, sequence tokens, sequence tags, the appearance of the word, special lists which related to the category, Viterbi algorithm, and some special sentence structure in the reviews. For instance, people like to put a lot of exclamation marks or emoji after a specific word to describe it which usually belongs to one category.

The fourth problem is in the testing phase. By entering a review to our model, we can generate its labels of the seven categories based on our trained model. Through the evaluation process, we can easily obtain Accuracy by comparing with the labeled test data. But, how do we compute meaningful Precision, Recall, and F-measure statistics to explain our data, since we have seven different categories and the OTHER category does not have positive and negative labels. Furthermore, according to the generated labels, how do we formulate a reasonable scoring equation to give rating of different categories.

System Workflow

The main work consists three parts. 1. Scrape review data from Yelp into .csv file and do the labeling after parsing .csv file into a standard format in .txt file. [1] 2. Build useful feature sets and train the model. 3. Test other reviews and compute different category's ratings.

In the data preparation phase, we scrape review data from Yelp by using Yelp Fusion API and python html package. Here is an example on how to run the python code getReview.py to extract Yelp's review from a given restaurant. Suppose there is a restaurant's url on Yelp: `https://www.yelp.com/biz/club-a-steakhouse-new-york`. First, you need to create a folder reviews at the same level of the python code. Then, the command is python getReview.py -b "club-a-steakhouse-new-york" -r "100". -b indicates certain business. -r means to get the most recent x reviews. If -r is omitted, it will scrape all the reviews of the indicated restaurant in a sorted order based the date. We chose three restaurant to label their 40-recent reviews which are "Peter Luger in Brooklyn", "Saigon Shack", and "Tim Ho Wan". In addition, when preparing the training and

[1] The standard format is that sentences in a review will be parsed into .txt where one word is in one line just like the format in WSJ corpus. We implement this .csv parser on our own. It's not perfect since there are some wrong grammars and strange words in the reviews, but most of the results are reasonable.

testing data, we decided to choose odd numbers of reviews as training data and even numbers of reviews as testing data. By doing so, we can reduce bias in data since the date for each testing and training data is close to each other.

In the feature building phase, we have tried these features with their effectiveness (gain/drop):

1. sequence POS features (gain),
2. POS combination like NN+VB+JJ (gain),
3. sequence token features (drop),
4. sequence tag features (drop),
5. the appearance of the token word (gain),
6. special lists which related to the category (gain),
7. special sentence structure in the reviews (small gain).

The idea of the sixth feature is referred from the assignment 6. In order to increase the accuracy and F-measure of our model, we create word lists for each type in the first sixth category. For example, for the "Taste" category, we search word "delicious" and "distasteful" on `Thesaurus.com` for all the synonyms and save the word list as two separate files. For other categories, we use "expensive" and "cheap" for "Price" category, "friendly" and "rude" for "Service" category, "clean" and "dirty" for "Sanitation" category, "convenient" and "inconvenient" for "Location" category. Then, we can use these lists as our dictionary when building features. As the result, the improvement of F-measure is obvious after using word lists. The training process follows the procedure in the assignment 5 and 6 which based on opennlp's MEMM.

In the testing phase, we have tried Viterbi tagging, but the improvement is quite limited. Then, we have computed Precision, Recall, F-measure, and Accuracy based on different (true positive/true negative/false positive/false negative) statistic. We have detail discussion in the following sections. Moreover, we have calculated different category's rating according to our system's tagging result. Additionally, we chose five more restaurants to test our rating system which are "Asian Jewels Seafood Restaurant

	Peter Luger	Saigon Shack	Tim Ho Wan
Accuracy	96.81	97.39	97.05

Table 1: Accuracy of the three restaurants

in Flushing", "Burger and Lobster", "Din Tai Fung in Seattle", "Pho Vietnam 87", and "Porter House Bar and Grill". In the discussion section, we provide figures to visualize our rating system and give a comparison with Yelp's rating system. The figure 1 is our system workflow.

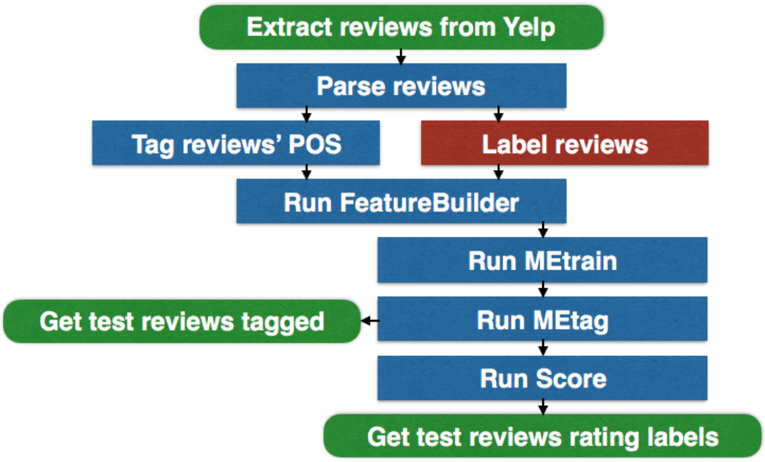

Figure 1: System Workflow. Green blocks stand for input/output. Red block is annotation process. Blue blocks are our implementation.

Data Analysis

First, we can easily calculate the accuracy of the response tags over the labeled data. Please find the accuracy in the table 1. Here, we can see that the accuracy is very high since most of the tags are OTHER. Therefore, we look inside the detail of each correct response tag out of the corresponding labeled tag and calculate its accuracy. For instance, for the restaurant Peter Luger, the number of correct FRIENDLY response tag is 7 and the number of FRIENDLY tag is 16. The accuracy is 43.75%. Similarly, for the restaurant Saigon Shack, the number of correct YUMMY response tag is 12 and the number of YUMMY tag is 22. The accuracy is 54.55%. Figure 2, 3, 4 shows the aforementioned detailed accuracy. Note that if the tag has no number, it is because there is no such tag for that restaurant.

As we can see from the result, label CLEAN, DIRTY, CONVENIENT and INCONVENIENT all have only few examples. We believe there are

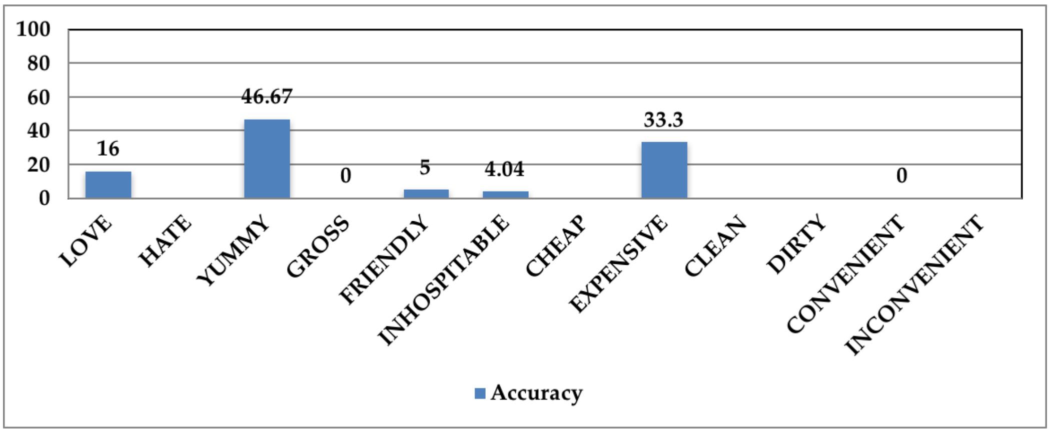

Figure 2: Tagging Accuracy of Peter Luger in Brooklyn

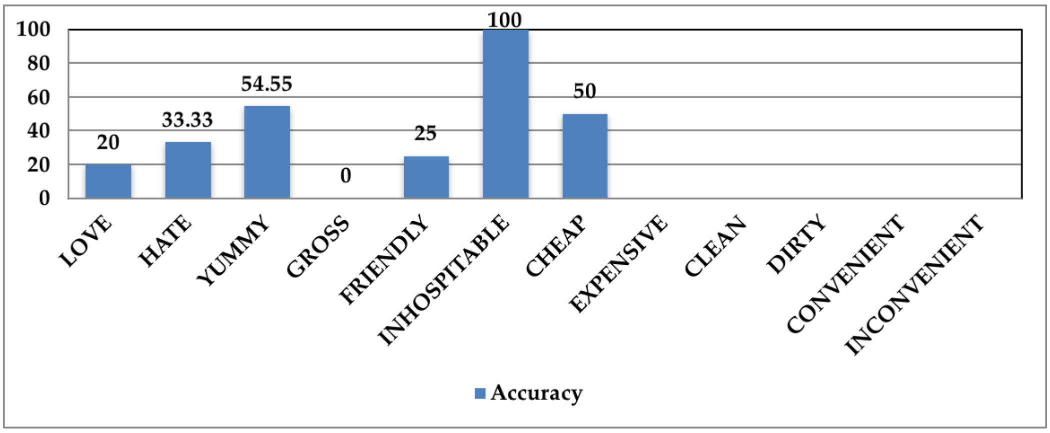

Figure 3: Tagging Accuracy of Saigon Shack

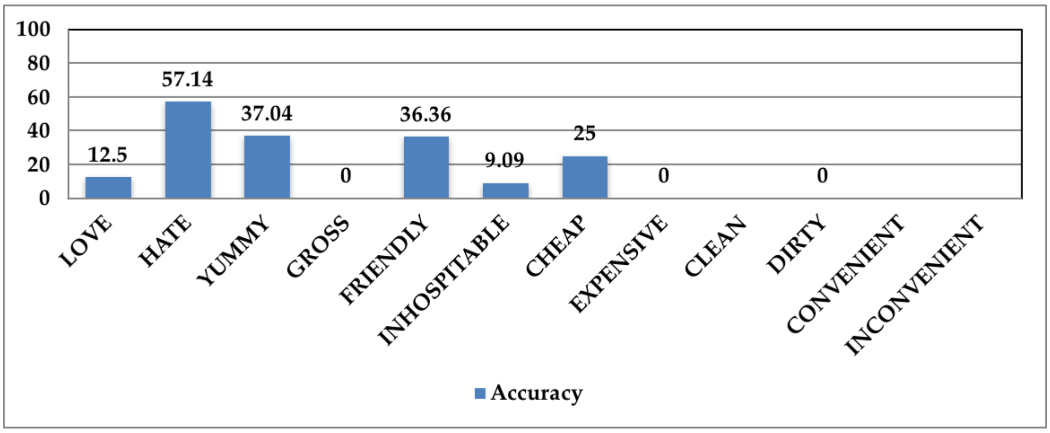

Figure 4: Tagging Accuracy of Tim Ho Wan

some explanations for this. For the location, since the restaurants we choose are all in Manhattan, it is possible that most of the people take train to these restaurants so they seldom mention about traffic in reviews. As for the sanitation, first, these restaurants all have clean environment. Second, cleanliness of a restaurant is regulated by government. If one restaurant is too dirty, authorities have the rights to shut down the business. Thus, restaurants need to keep clean in order to meet the requirements.

Discussion and Challenges

During tagging process, it is very hard to have perfect data since some users may have typos or grammar mistakes in their reviews. Besides, some of them use words with sarcasm so it is difficult to get the real meaning behind the metaphor. Some of them use emoji or irregular emphasis to describe a restaurant such as ”yummyyy” or ”baaaaad”. It is not easy to find the corresponding words with these special words, but they are invaluable for extracting information from reviews. Moreover, some of the users talk about other restaurants in their reviews so we could not find related information for this restaurant.

Another challenge is that some reviews may contain foreign languages other than English. For example, in a Spanish restaurant, a lot of users used ”delicioso” instead of ”delicious”. For some dim sum restaurants, some customers may use Mandarin when writing reviews. Therefore, we try to avoid this defect by choosing American restaurant or restaurants which sell American food. Lastly, some adjectives and nouns are too general, they can describe different categories. Take good/bad and high/low for example. They can both be used for taste and price. Hence, it is hard to tag them into a specific category.

During evaluation process, when calculating accuracy, it is challenging to use confusion matrix since we have more than one category. In order to explain and well illustrate our result and model, we spend a lot of time to define a reasonable formula for accuracy, recall and precision.

First, we only choose tags from testing and training data that belong to the same category. By doing this, we can get a binary classification. Then we can calculate true positive, true negative, false positive and false negative in confusion matrix for each category. Take category ”Price” for example, we only pick testing and training tags are either CHEAP or EXPENSIVE, then we put:

- Training and testing tags that both equal CHEAP as true positive.
- Training and testing tags that both equal EXPENSIVE as true negative.
- Training tag equals CHEAP and testing tag equals EXPENSIVE as false negative.
- Training tag equals EXPENSIVE and testing tag equals CHEAP as false positive.

Then, we can compute F-measure for certain category. For the tagging accuracy. The accuracy result is very high because there are many OTHER in our tagging and once the word was tagged, it was nearly 100% correct.

After that, we categorize reviews and determine rating for every category. By calculating the sum of one type tag then divided by the sum of all tags in one category. For example, 20 tags from 40 reviews were tagged to LOVE in "Overall" category, 10 tags from 40 reviews were tagged to HATE in "Overall" category, the rating for "Overall" would be 20/(20+10) * 5 = 3.33. We then had 6 ratings for each category for every restaurant. By aligning these ratings with the rating from Yelp, we can have a much clear picture of the detail of the Yelp rating. This provides an effective and useful information for business owners. In Figure 5, the "Price" category is EXPENSIVE with rating 3.33, this means there are 2 over 3 people thought Peter Luger was expensive and the other person thought it was cheap. In Figure 6 and 7, both "Sanitation" and "Location" have no ratings. This is because the recent forty reviews from both restaurants did not mention anything about convenience nor cleanliness. At the end of the article, we provide more alignment results between our rating system and Yelp's rating in the Figure 8, 9, 10, 11, 12.

Conclusion and Future Work

As from the figures, we can see the average ratings from these six categories is similar with the rating from Yelp. We believe our rating system can help business owners to have a better understanding about the reviews from Yelp. Still, there are some on-going enhancements of this rating model. In the future, we have three big directions to work:

1. Add sentiment analysis, so we can know which word is more likely to become positive or negative.

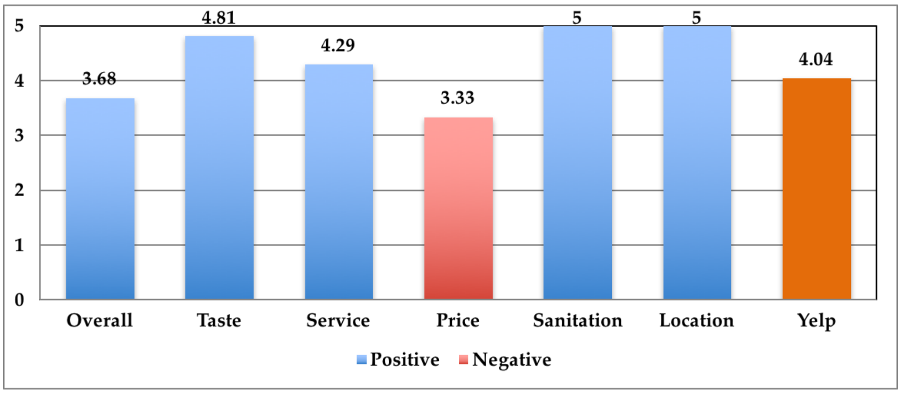

Figure 5: Result of Peter Luger in Brooklyn

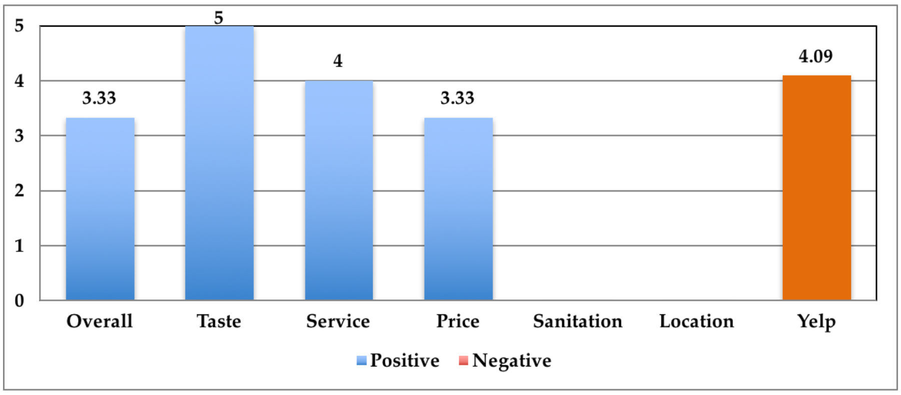

Figure 6: Result of Saigon Shack

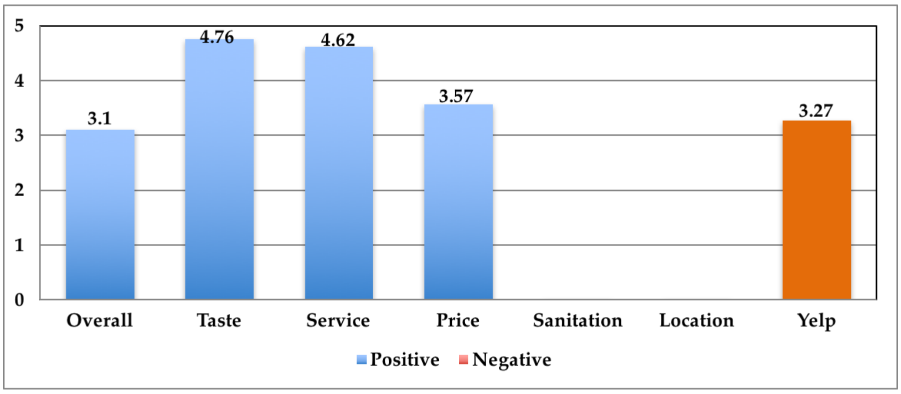

Figure 7: Result of Tim Ho Wan

2. Improve dictionary list by adding words with different POS.
3. Label more reviews so we will have more data to train our model.

With all these improvements, we believe that they will help our systems to become more accurate and reliable.

Workload Distribution

Ching-Hsiang Hsu

1. Implement ParseCSV.java to parse reviews
2. Label 60 reviews of three different restaurant
3. Implement FeatureBuilder.java
4. Implement MEtag.java
5. Implement Score.java
6. Write report

Li-Hao Liu

1. Implement getReview.py to extract reviews from Yelp
2. Tag reviews' POS
3. Label 60 reviews of three different restaurant
4. Define F-measure computation and scoring function
5. Write report

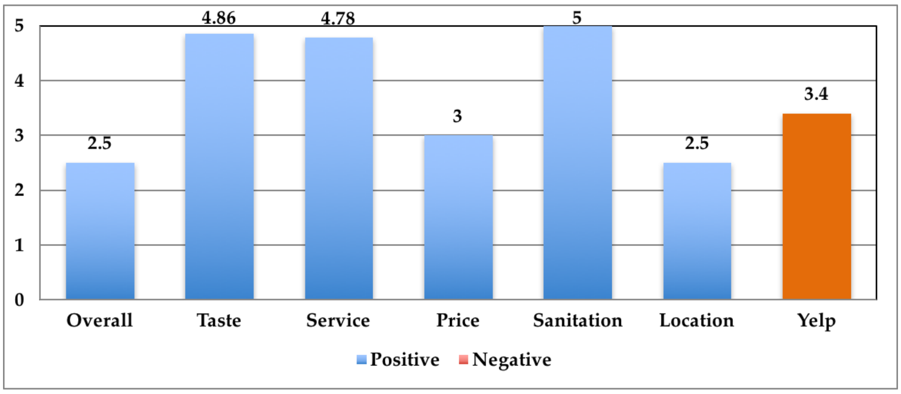

Figure 8: Result of Asian Jewels Seafood Restaurant in Flushing

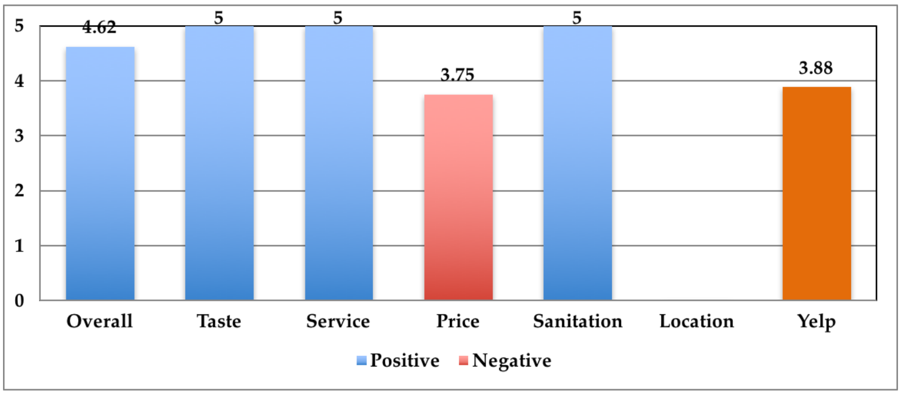

Figure 9: Result of Burger and Lobster

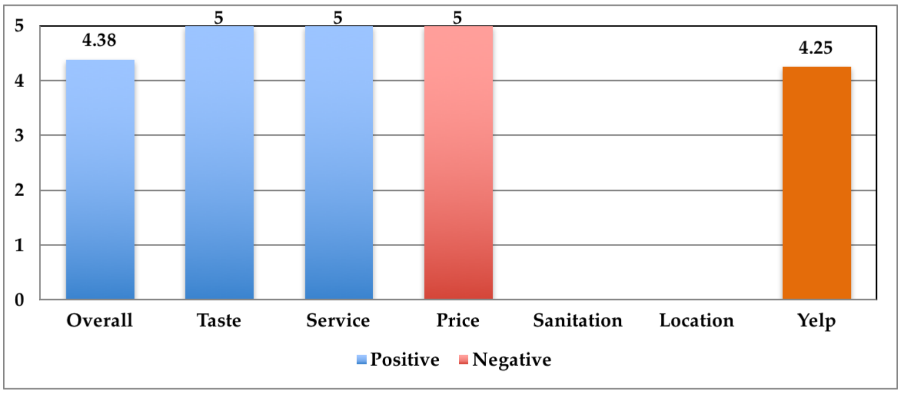

Figure 10: Result of Din Tai Fung in Seattle

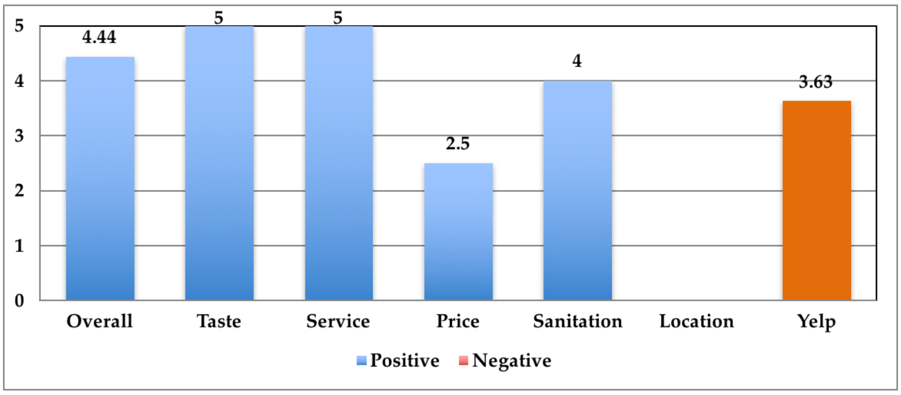

Figure 11: Result of Pho Vietnam 87

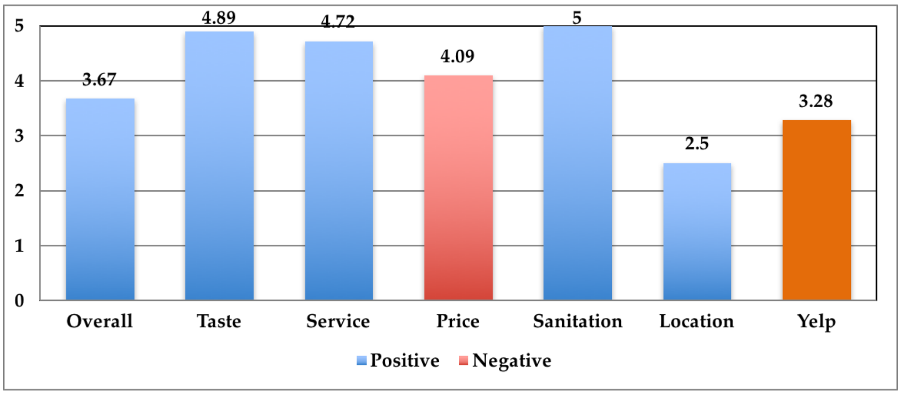

Figure 12: Result of Porter House Bar and Grill