Yelp Review Annotation for Machine Learning

Qishen "Justin" Su qsu@brandeis.edu
Kelley Lynch kmlynch@brandeis.edu
Yuanyuan Ma yyma@brandeis.edu

Abstract

Yelp restaurant reviews are helpful and useful for customers to understand the quality of the food and service in a given restaurant. For a general understanding of the quality of food and service of restaurants, a star rating for restaurants are given. In this course project, we proposed to provide classification for each food item of a restaurant, which could help customers make a better decision on what to order. In this paper, we described our annotation task and results, as well as certain machine learning experiments that we have performed.

1 Introduction

Yelp provides an opportunity for customers to review services and is the go-to site for restaurant recommendations. With general summaries including star rating(from 1 to 5, with 5 being the best), pricing, and restaurant categories, one can easily find information about a particular restaurant. Despite all the information that is available, one still needs to read through the reviews to discover the quality of a food item offered by that restaurant. The volume of comments on a particular food buried in the reviews makes it difficult for a user to extract information and make an educated choice. Mining information on the quality of a particular menu item, therefore, would be a time-saving addition for customers.

There has been a trend in automatically summarizing reviews and providing results for easy processing (Blair-Goldensohn & Hannan, 2008). Among the different techniques, supervised machine learning, which has been widely adopted for sentiment analysis(Hu & Liu, 2004), requires a large amount of annotated data. The creation of annotated data is often less focused than machine learning approaches. In this study, we developed a gold-standard annotation guideline for annotators to mark mentions of food, quality, anaphora with correct and suitable tags that a machine learning algorithm can take in as input(Pustejovsky & Stubbs, 2012). Experiments with Support Vector Machine (SVM) show that the annotation improves the performance of sentiment analysis over food items for a particular restaurant.

In creating the guidelines, we focused on the annotation of a particular category of cuisine, Mexican food. Mexican food is eaten regularly by more than half of the people in a survey conducted by National Restaurant Association(National American Restaurant, 2015). From Yelp's Challenge dataset, Mexican restaurants rank in the top 5 most reviewed categories (Figure 1).

Restaurant Category	Number of Reviews
Nightlife	212,618
Bars	205,693
American (New)	187,984
American (Traditional)	182,300
Mexican	144,727

Figure 1: Top Five Most Reviewed Restaurant Categories

The task of sentiment analysis for food items is composed of sub-problems like named entity recognition, ontology matching, and coreference resolution. The corpus created by the guidelines provided tags and highlights of food items, descriptions of quality, anaphora and the ontology of ingredients. They can be applied to different machine learning sub-tasks to develop a classification system for Mexican foods.

2 Yelp Review Annotation

2.1 Corpus

Our corpus consisted of restaurant reviews provided by Yelp during round 7 of the annual Yelp Dataset Challenge. Reviews from 4 Mexican restaurants in Phoenix, Arizona were annotated by 3 annotators. The annotators were instructed to ignore reviews that did not mention any menu item. Many reviews describe only the service or atmosphere of the restaurant and those types of reviews are outside the scope of this project. The following chart shows the number of reviews about each restaurant that were given to annotators, the number of reviews that were included in the gold standard corpus, and the average star rating for each restaurant.

	Annotation Package	**Gold Standard**	**Star Rating**
Calico Jack's	143	72	1.5
Carlos O'Brien's	120	84	3.4
Mi Amigo's	115	76	3.0
Roberto's	166	113	4.0

Figure 2: Review Counts

Each review had in its metadata a star rating assigned by the writer of the review. These star ratings are averaged by Yelp to get the restaurant star rating.

Star Rating	**Number of Reviews**
1	68
2	52
3	64
4	80
5	81

Figure 3: Reviews per Star Rating

Our gold standard corpus consisted of approximately 41,119 words. The reviews varied greatly in length, but the average length was 120 words.

2.2 Data Collection

The dataset is in json format, which allowed us to query relevant reviews that we needed without any difficulties, i.e. we were able to obtain reviews from Mexican restaurants and restaurants of certain star ratings. Then, we used a list of Mexican food names to further filter the reviews, to only retain reviews that mention actual food items.

Tag	**Count**
Food	1417
Anaphora	336
Quality	1421
Part-of	385
Opinion	1543
Coreference	500

Figure 4: Gold Standard Corpus Tag Counts

Extent Tags	**Linked Tags**
FOOD	PART_OF
QUALITY	OPINION
ANAPHOR	COREFERENCE

Figure 5: Annotation Tags

2.3 Annotation Guidelines

The annotation guideline for this project provided instructions for tagging reviews so that the reviews and the features within the text of the reviews could be used as input for machine learning algorithms. The tags are grouped into extent tags and linked tags (Figure 4).

(1) **FOOD** is an extent tag that spans over texts mentioning a food item. It excludes articles such as "the" and "a", and includes the prepositional phrases headed by "with" or "con" (such as "carne asada potato with cheese" or "chile con queso"). These phrases are included because they are established parts of a dish's name. For the purpose of sentiment analysis, we exclude the food items that do not have any description on their qualities.

(2)**QUALITY** marks the quality of a food item. It has an attribute of being "positive", "negative" or "neutral". Descriptions on prices are tagged as quality. Phrases related to food items that do not convey sentiment judgment are given the attribute "neutral".

(3)**ANAPHORA** highlights all mentions of a food item that has appeared previously. It could be pronouns or food names. For quantifiers such as "everything", the annotators should make judgment according to the context.

(4)**PART_OF** indicates two relationships. One is from ingredient to the food such as "beans"

to "burrito", or from food to a combo, such as "chips" to "chips and salsa". This semantic relation derives from the fact that Mexican food are often different combinations of the same ingredients.
(5)**OPINION** links quality with the foods it describes. It has an attribute being "explicit" or "implicit". Sarcastic comments indicated by quotation marks should be labeled as "implicit" opinions.
(6)**COREFERENCE** connects later mentions of a food item to its first mention in the review. Any quality description in the sentence that the anaphor appears, should be directed to the anaphor instead of the first mention of a food.

An example including all tags would be "The $carnitas\ taco_{food}$ was $SO\ good_{quality^+}$. $It_{anaphor}$ comes on a $soft\ shell\ taco_{food}$ with $pico\ de\ gallo_{food}$ and $guacamole_{food}$". The detailed annotation specification can be found in our annotation guideline.

2.4 Annotation Process

The reviews were annotated over 4 weeks. The first week's annotation package consisted of 143 reviews about the restaurant Calico Jack's. Seventy-two of the reviews in the package contained a description of a food item and were included in the Gold Standard corpus.

Following the first week of annotation, changes were made to the annotation guidelines to provide more examples of the use of anaphora tags. Specifically, examples were added to demonstrate tagging words such as "both", that serve as anaphors for multiple food items. No further changes were made to the guidelines following this revision. Over the next three weeks, the annotations were completed for Carlos O'Brien's, Mi Amigo's, and Roberto's, in that order. Each restaurant's reviews took one week to annotate.

2.5 Results

We evaluated the results of the annotation using Krippendorff's Alpha.(Figure 6) We chose to use Krippendorff's Alpha because we had 3 sets of annotated data and it was particularly important for us to measure consistency in the span of extent tags. The span of extent tags was important, because in Mexican food names, the name of one menu item can have within it, the name of another menu item, e.g. "quesadilla plate" and "chicken quesadilla plate".

Extent Tags	Inter-Annotator Agreement
Food	0.4394
Quality	0.5306
Quality:type	0.6780
Anaphora	0.4394
Cross-tag	0.5562

Figure 6: Inter-Annotator Agreement on Extent Tags

3 Machine Learning Experiments

3.1 Experiment Pipeline

We designed a pipeline for our experiment, though not every step was achieved or attempted due to constraint of time and data set size. However, it is important to understand how eventually each part works together. *Name entity recognition* identifies all the food items in a review, and then *reference resolution* finds all the descriptions that refer to the same food item, and *food ontology* finds all ingredients of each dish.

The next step is to extract all the descriptions regarding each food item, and then *sentiment analysis* is performed on all the descriptions of the food item in order to identify the overall positiveness and negativeness of each food item.

In the following experiments, we were able to experiment with named entity recognition and sentiment analysis. In the future, we would like to experiment on coreference resolution and food ontology learning.

3.2 Name Entity Recognition

In order to determine the polarity and characteristics of a menu item based on the customer reviews, one of the important and essential parts is to correctly identify the food names in a review, so as to locate the informative parts and sentences of the review.

Our first attempt to extract menu items names from reviews used the Wikipedia list of Mexican Foods. We used a regular expression to find items in the reviews that are included in the Wikipedia list. This method performed poorly for multiple reasons. First, many restaurants have unique

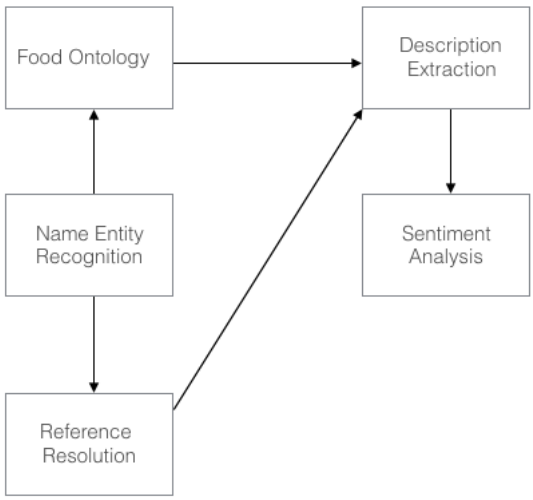

Figure 7: Experiment Pipeline

menu item names. For example, one item in our reviews was called the "Blue Margarita". This method would only match "Margarita". Other menu item names contain multiple foods. For example, with this method the food name, "Al Pastor tacos" would be matched as "al pastor" and "tacos".

We used Micro-Precision, Micro-Recall, and Micro-F-score to evaluate the baseline. We used these metrics because the lengths of reviews and the number of menu items in a review vary greatly.

Micro-Precision	36.9%
Micro-Recall	30.6%
Micro-F-score	33.3%

Figure 8: Baseline Menu Item Name Extraction

Next, we utilized the Stanford Name Entity Recognizer (Stanford NER) (Finkel et al., 2005), which is an implementation of conditional random field (CRF) sequence models, to identify the food names in a review, so as to correctly determine the informative part of the review and obtain useful information about the food mentioned in the review. Each FOOD tag in the gold standard corpus is used to locate the food names in the corpus, and generated a new data set where each noun phrase that is food name is tagged with FOOD tag, and other tokens are tagged with the value 0 (see Figure 4). The training data set contained 309 files, and test set contained 35 files. Figure 16 contains the features used in training the NER model when it provides the best results, and the descriptions in the table are from the Stanford JavaNLP API Documentation (Stanford NLP Group, 2005).

My/0 friend/0 had/0 the/0 Chicken/FOOD Quesadilla/FOOD.

Figure 9: A Sample of Data Set Used for NER

The performance the Stanford NER on identifying food names is the following:

Precision	**Recall**	**F1**
0.7857	0.7971	0.7914

Figure 10: NER Performance

With the small amount of training data, the performance of the NER is satisfactory. We believe if there were more annotated data, the performance would probably be better.

3.3 Sentiment Analysis

With a set of descriptions for a given dish, our goal is to determine which of them is positive or negative, and thus sentiment analysis is another core component to this project.

In order to evaluate the results, we extracted each distinct food name in a restaurant, as well as its descriptions. To make evaluation simpler, we decided that a positive description adds 1 point, a negative one subtracts 1 point. If the sum of all points is greater than 0, then the overall quality of a food item is positive; and if that is less than 0, then the quality is negative. This method may be insufficient, as certain positive or negative polarity sentences carry more weight than others, however we did not have the time or data necessary to account for this. For the following experiments, we compare our classification results against the point value of the food items for evaluation.

Our baseline for sentiment analysis used the food list extracted from the reviews by the NER baseline. We trained a Naive Bayes Classifier using a training set of reviews for each restaurant and the star rating meta-data for those reviews. This method was successful for 3 of the 4 restaurants, however, the success of the classifiers was

due to overfitting which can be seen by comparing the performance of the classifiers for restaurants with a particularly high or low average rating to the performance for the restaurant with the most neutral average rating (Figure 8).

Restaurant	Positive	Negative	Accuracy
Calico Jack's	0	10	100%
Carlos O'Brien's	14	0	78.6%
Mi Amigo's	7	11	38.9%
Roberto's	25	0	80%

Figure 11: Baseline Classification Results per Restaurant

For 3 of the 4 restaurants, the classifier assigned all foods the same rating. For the 2 restaurants whose average star rating was greater than 3, all foods were classified as positive and for the one restaurant whose average star rating was less than 3, all foods were classified as negative. The classifier for the restaurant whose average rating was 3 stars labeled classified foods as both positive and negative with poor performance. The three classifiers that performed well were unable to accurately classify any of the foods whose correct classification differed from the restaurant average. This suggests that the classifiers were overfit to the data.

This method is inadequate because the star rating meta-data for a review does not correspond well to the sentiment of all of the words in the review. For instance, in one review the description of the food was very positive with the exception of one sentence, saying that reviewer got food poisoning from the food. Most of the sentences of the review had a positive sentiment, but it was a 1 star review. Training the classifier on sparse data, some of which had these types of problems, made the classifier ineffective.

We applied the classifier to food descriptions by assuming that if a food is mentioned in a particular sentence, then that sentence is about that food. In addition to the sentences where a food name appears, we also assumed that the following sentence would be about that food unless the following sentence mentioned a different menu item. We decided to include the sentence following the mention of a menu item because reviewers often used one sentence to identify a menu item and the following sentence to describe it while using an anaphor to refer to the item.

Although in our annotation, "neutral" is one of the types for opinion, besides "positive" and "negative", QUALITY tags with "neutral" type are excluded, because our objective for sentiment analysis is to only display a positive or negative description of a dish. Due to the small size of our data set, we decided to use SVM^{light} implemented by Thorsten Joachims (Joachims, 1991) to classify sentiment. We extracted all the texts in the QUALITY tags and their polarity, i.e. positive or negative, and then use the words as features to train on a model. 80% of the data were used for training, and 20% were for testing. Additionally, we also trained a model for quality tags that are of explicit opinion. The results in Figure 12 are somewhat satisfactory.

	Precision	Recall	F1
Explicit and Implicit Opinions	83.78%	83.78%	83.78%
Explicit Opinions	80.23%	95.83%	87.63%

Figure 12: SVM^{light} Classification Results for Sentiment Analysis

Next, we applied the model to classify the descriptions for each unique dish of each restaurant. The following are the number of food items that the classifier assigned as positive or negative, and the accuracy of the classifier for each restaurant. The results in Figure 13 shows improvement in accuracy compared to those in Figure 11.

Finally, we amalgamated all the descriptions for each food item of each business, and then used the SVM model above to classify the overall positiveness and negativeness of the food items. The results are shown in Figure 15. Overall, the performance of SVM^{light} is satisfactory for sentiment analysis. d

Restaurant	Positive	Negative	Accuracy
Calico Jack's	2	10	91.67%
Carlos O'Brien's	15	2	92.47%
Mi Amigo's	14	3	88.24%
Reberto's	21	3	87.50%

Figure 13: SVM^{light} Classification Results for Each Restaurant – Explicit and Implicit Opinions

Restaurant	Positive	Negative	Accuracy
Calico Jack's	1	7	100%
Carlos O'Brien's	17	1	100%
Mi Amigo's	10	3	92.31%
Reberto's	15	2	88.24%

Figure 14: SVM^{light} Classification Results for Each Restaurant – Explicit Opinions

	Precision	Recall	F1
Explicit and Implicit Opinions	91.89%	87.18%	89.47%
Explicit Opinions	96.97%	94.12%	95.52%

Figure 15: SVM^{light} Classification Results for Each Food Item

4 Future Work

4.1 Food Ontology Learning

Often customers do not only describe a dish as a whole, but also describe its ingredients. For example, a customer may describe the beans and rice in a burrito, and the quality of the beans and rice are one component of the quality of the burrito. Therefore, it is also important to identify the ontology of a dish, i.e. to determine what is part of a dish and what is an actual dish, and then find out all the related descriptions of a dish.

There are two possible approaches to this issue: logical learning and lexical learning (Lehmann & Voelker, 2014). Logical learning "derive[s] ontologies from structured knowledge", which is mainly rule-based, and with lexical learning, "[o]ntology learning from natural language text based on information extraction and text mining techniques"(Lehmann & Voelker, 2014).

However, since our current data set is too small and data are sparse, with the average frequency of 2 for a food name being mentioned, we were unable to build meaningful model to identify food ontology by using any probabilistic model.

One possible way to deal with the issue of data sparseness is to extract recipes from websites like Food.com, where the meronyms of a dish can be extracted, and we can use the data to build a statistical model on meronymy of food.

4.2 Coreference Resolution

Coreference resolution attempts to find all expressions that refer to the same entity (Stanford Coreference, 2010). In our case, we wish to find all the mentions of a particular food item. The importance of coreference resolution lies in the fact that it allows us to pin-point the scope and target of sentiment expressions (Blair-Goldensohn & Hannan, 2008). An example where coreference resolution should be applied is "After trying to find some good queso$_{food}$ in the Valley with very little success, I finally found the good stuff at Carlos O'Brien's. It$_{anaphor}$ was delicious."

We experimented with a state-of-the-art coreference resolution system, however the results showed that this system was not adequate for our purpose (Stanford Coreference, 2010). The precision and recall of the system were 2.94% and 24.39% respectively. Mentions of places, people, dates, and restaurants were found, and since these items were not part of our project, which caused the precision to be quite low. A considerable amount of anaphora in the form of partial food names are not recognized, thus the low recall. Given this as a baseline, it is likely that we could significantly improve the results by feeding our annotated data to a learning algorithm.

Studies on coreference resolution provide possible solutions to this problem. While some solutions heavily rely on results of named entity recognition and dependency parsing (Stanford Coreference, 2010), one possible approach we can adopt in our project is the cluster-ranking model (Rahman & Ng, 2009). It combines results from entity-

mention model and mention-ranking model from an SVM ranker-learning algorithm.

5 Conclusion

Overall, some more improvement and specification on our annotation guidelines are necessary to further improve the quality of the data that we can use for machine learning. With that said, the quality of the gold standard corpus allowed us to achieve promising results in the experiments. We are confident that if there were more data, the results of the experiments would be further improved. Moreover, a larger amount of data could enable us to experiment on ontology learning and coreference resolution. We believe that eventually, when every part comes together, classification of each food item would be helpful and informative for customers to make a better decision on ordering food.

Acknowledgments

This machine learning project could not have been attempted without the tireless efforts of our annotators, Xinhao Wang, Jose Ramirez, and Matthew Garber. The quality and commitment of their hard work is much appreciated.

References

T. Joachims 1999. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, MIT-Press, 1999.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370.

Stanford NLP Group. Class NERFeatureFactory. *Stanford JavaNLP API Documentation*, http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html.

Jens Lehmann, and Johanna Voelker (Eds.) 2014. Perspectives On Ontology Learning. *Studies in the Semantic Web AKA / IOS Press*. pp. xiii-xiv.

Sasha Blair-Goldensohn, and Kerry Hannan 2008 Buidling a Summerizer for Local Service Reviews *NLPIX 2008*

National Restaurant Association 2015 Global Palates: Ethnic Cuisines and Flavors in America *Global Palates 2015*

Altaf Rahman and Vincent Ng 2009 Supervised Models for Coreference Resolution *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 968977*

James Pustejovsky and Amber Stubbs 2012 Natural Language Annotation for Machine Learning *OReilly Media, Inc.,*

Minqing Hu and Bing Liu 2004 Mining and Summarizing Customer Reviews *2004 ACM*

Stanford NLP Group Deterministic Coreference Resolution System *Stanford CoreNLP*, http://nlp.stanford.edu/software/dcoref.shtml

Parameter	Value	Description
useClassFeature	true	Include a feature for the class (as a class marginal). Puts a prior on the classes which is equivalent to how often the feature appeared in the training data.
useWord	true	Gives you feature for w
useNGrams	true	Make features from letter n-grams, i.e., substrings of the word
noMidNGrams	true	Do not include character n-gram features for n-grams that contain neither the beginning or end of the word
maxNGramLeng	5	If this number is positive, n-grams above this size will not be used in the model
usePrev	true	Gives you feature for (pw,c), and together with other options enables other previous features, such as (pt,c) (with useTags)
useNext	true	Gives you feature for (nw,c), and together with other options enables other next features, such as (nt,c) (with useTags)
useSequences	true	Does not use any class combination features if this is false
usePrevSequences	true	Does not use any class combination features using previous classes if this is false
maxLeft	4	The number of things to the left that have to be cached to run the Viterbi algorithm: the maximum context of class features used.
useTypeSeqs	true	Use basic zeroeth order word shape features.
useTypeSeqs2	true	Add additional first and second order word shape features.
useTypeySequences	true	Some first order word shape patterns.
useDisjunctive	true	Include in features giving disjunctions of words anywhere in the left or right disjunctionWidth words (preserving direction but not position)
useWordPairs	true	Gives you features for (pw, w, c) and (w, nw, c)
useNextSequences	true	Does not use any class combination features using next classes if this is false
useLongSequences	true	Use plain higher-order state sequences out to minimum of length or maxLeft
maxRight	4	The number of things to the right that have to be cached to run the Viterbi algorithm: the maximum context of class features used.

Figure 16: Parameters used in training NER model