# Yelp Review Context Classifier

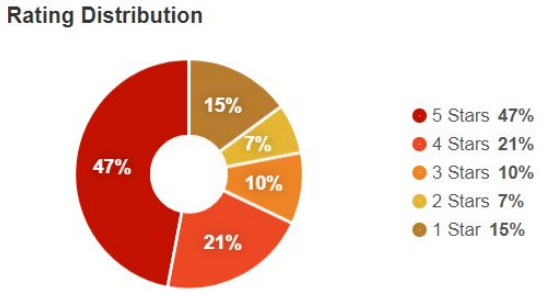
Team 47: Paul Moreno, Ilean Monterrubio

## Abstract

Vertical search engines have become increasingly popular, they sift through limited databases for information. In particular location based mobile searching is extensively used for searching businesses that are close to the user or have high user ratings. Broad web searches cannot accommodate all of the user's searches. When it comes to specific topics with implicit assumptions are not captured in general web searching. One major example is Yelp, its content is specialized for users browsing information on service or goods. General web searches focus on a broad range of web page results while mobile searches focuses the results closer to the users location. Results from services like Yelp allow a user to rank results by distance ,ratings, or reviews. Customer preferences can be tracked by looking at the user's reviews of visits on yelp. Being able to track users experiences at a business could lead to greater customer traffic and sales for the business.
The software tool we have built allows businesses to classify their reviews and filter them to see if they are lacking certain areas. We chose to focus on analyzing if the customers based their reviews on goods or services provided by the business. Understanding the areas of opportunity and growth for a business is essential to maintain growth and a good client base.

## Introduction

The rise in popularity of mobile devices has led to search algorithms that implement data from the user's location and movements. Yelp and others leverage information available concerning the user when implementing searches using mobile devices. Users often use services like Yelp to search for new businesses to visit or to check for best user experience from reviews. Vertical search engines such as Yelp address the need for deeper more specific more relevant search results from a certain domain. The application of algorithms that leverage domain knowledge and implicit user specific information from the task increase the chances of the user receiving relevant search results.
The results yielded from Yelp can help drive users to certain businesses. This makes business reviews on Yelp a critical component of user feedback. According to a Harvard Business School case study, every star in a review leads to a 5-9% additional revenue. The star and the review are important to businesses seeking to maximize business from Yelp users. The importance of the star rating can help businesses track their performance as well as identify areas of weakness. Tracking star data can also lead to improved business performance and provide feedback on customer needs. Addressing customer needs is critical to many service based business.




Most of the apps available that leverage Yelp data track the customer habits, needs, or requests. They are centered in providing better search results and service to the users of Yelp. Our software in contrast provides the business user with categorized feedback that they can quickly track and use to target goals.
In this software tool, we choose to identify Yelp reviews according to whether the review is

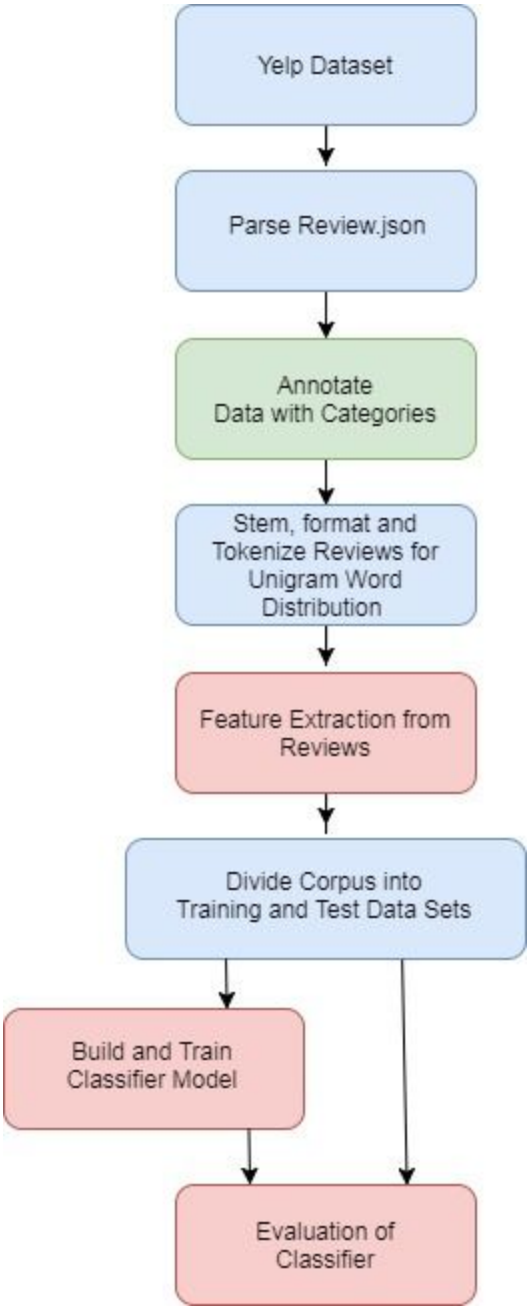
focused on the service provided or goods sold at the business. By tracking these  two topics a business can track their business strengths and weaknesses. In a restaurant, for example, it is important to have good service and good food. If one is lagging far behind the other the business could suffer. It also allows management insight into worker interactions with the customer. Poor customer service can be tracked and correlated to a hiring or firing of an employee. It is also a window into worker morale.

## Background

We started the project exploring the Yelp Fusion API. It provides users the ability to actively search Yelp and all of its businesses. The Business API only returns up to 3 reviews per business query. This was not enough to carry out our project and provide the training data necessary to do the topic and sentiment analysis of the yelp reviews. The amount of data to do our analysis was available from the Yelp Dataset Challenge. The data is open source and available to everyone.

## Implementation

The project was broken down into several tasks to better understand the tool desired. At a very early discussion with the team, it was determine this would be a binary classification problem. Each comment from the dataset fits in one ot the two classifications. A comment that classified as “goods” referred to a review centered around the quality of the product, whether it is food, coffee, clothing. The second classified category is “service” these reviews are centered around the service the customer has received at a restaurant, dry cleaners, movie theaters.




### Data Implementation

The first task to building the tool was to collect the data and review its format. Yelp provides access to a collection of datasets through their Yelp Open Dataset program two different formats, JSON or sql file. We had trouble downloading and opening the files due to their size and their file types. The JSON file was too large for a text editor and the sql file required

MySQL for editing and viewing. We wanted to leverage the labs from class and use the python commands we learned. Python has libraries to use sqlite3 but not for MySQL which steered us away from that format. We decided to go with the JSON format option since it allowed direct access to the data and allowed us to load the data into OpenRefine. Upon inspection of the data downloaded we notice there were several files focusing on different areas, the only file of interest for the project was the review.json file. OpenRefine was used to do initial cleaning, parse,and view the columns of interest most importantly the review format. We also used OpenRefine because it is capable of loading the large amount of data.

In order, to easily manipulate the data the file was converted to comma delimited file (csv). This made the data into a table. Yelp provides through their Github repository a python script that converts the file from json to csv. We modified to script to output a manageable amount of reviews that we could annotate. This also allowed us to see the format of the data. The header of the file was as follows: business_id, date, review_id, stars,text, type, user_id, cool, useful, funny. The next decision was to determine how the tool would be used. The tool would be used by individual businesses only concerned with the comments they have received. This realization led to treating the entire file as a single business, leaving us to focus on the classification of the reviews. Each review is treated as a document to be classified as service or good.

During the Text Information course we used metapy library while it is powerful, it lacks documentation and deterred us from using the library. The choice to use Natural Language Toolkit (NLTK) was made as a group based on its extensive documentation and examples using python.

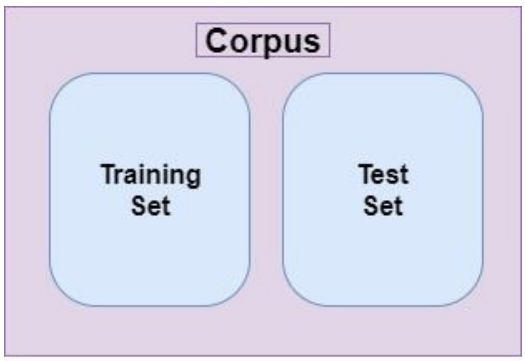
## Tokenization

The data in the csv file is loaded into memory and tokenized to extract the word distribution to selecting relevant features for the learning method. We start with using the “Bag of Words” approach and cleaning all of the data using NLTK which was a challenge and required some trial and error to get the best results .First we tokenize a 1,000 comments. Second we made all the words lowercase to better pull them together for removing all the stop words. We found that the built-in stop words distribution was not sufficient for our classification analysis. It was not as complete as the one used in class since it left many punctuations and special characters not needed. We imported the stopword.txt file used for the machine problem from class. Lastly, we stemmed the word distribution using the PorterStemmer built into NLTK and worked as desired. After cleaning and stemming the comments, the words were collected from the most frequent to the least frequent in a file called popular1.txt.

## Review Classification

We used Naive-Bayes and LinearSVC classifiers to detect patterns in our corpus that would identify the categories of our reviews. The classifier is supervised since we are using some of our data as training data. The corpus of 400 reviews is used to build and test the review classifier for the software. The reviews have been labeled with categories manually. In labeling the reviews for classification we tried our best to have uniform distributions of our two categories in both the training and test data sets. The set of first 225 reviews was used as training data for the classifier, that automatically tag s new documents with the appropriate category labels. The remaining 175 reviews were used to test the classifier and calculate quantitative scores regarding accuracy. We limit the the number of simple unigram word features to the most popular words found during tokenization and saved in popular.txt. Using the first 1000 words from the popular1.txt file as features, each of the 400 comments were tokenized and then using the 1000 words to find which of the popular1 words are in each individual comment. The feature extractor check to see if each of the words is in the given review.

Doing this the NLTK will make the bridge to which words are talking about service or goods. In order to get Precision, Recall, and F1 measure each of the 400 comments needed to be pre tagged by the team. Using OpenRefine to add a new column labeled topic, each of us tagged 200 each those comments as either good or service. As we did this, we noticed many of the comments did mention service or goods, but there were a great number of comments that referenced both equally, but each of these comments we made the choice based on the on which they praised more.




Two different classifiers were used, NLTK Naives Bayes classifier and scikit-learn LinearSVC classifier. LinearSVC also known as linear support vector classification, is a Support Vector Machine classifier using a linear approached.

## Classification Evaluation

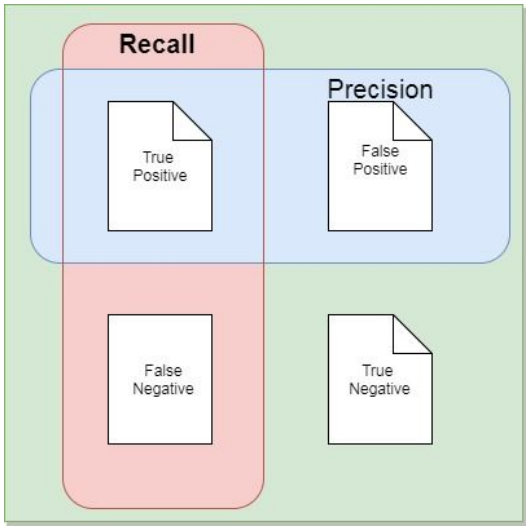
For evaluating that each classification model is accurately capturing word patterns in the comments, we evaluate the model with the annotated test reviews. The result of this evaluation is important to calculating the effectiveness of the training data. Evaluation can also be an effective tool for guiding us in future improvements to the model. It may be useful for the model to add more categories in the future since we noticed that there are reviews that have an equal amount of service words and goods words.

The script took each review in the test data bank and labeled it service or good using the trained classifier. The new data contained the label decided by a person, the review, the classifier label. We used the Naive Bayes Classifier in NLTK and the LinearSVC in scikit-learn libraries respectively to calculate the precision, recall, and Fmeasure for each category. This data was used to calculate each of the accuracy parameters.

Shown below the Precision, Recall, and Fmeasure for each category:
('Naive Bayes Classifier accuracy: ', 0.68)
('Precision of goods:', 0.3625)
('Recall of goods:', 0.7467811158798283)
('Fmeasure of goods:', 0.4880785413744741)
('Precision of service:', 0.33059548254620125)
('Recall of service:', 0.7666666666666667)
('Fmeasure of service:', 0.4619799139167862)

('LinearSVC Classifier accuracy:', 0.6571428571428571)
('Precision of goods:', 0.33689024390243905)
('Recall of goods:', 0.9484978540772532)
('Fmeasure of goods:', 0.4971878515185602)
('Precision of service:', 0.30579964850615116)
('Recall of service:', 0.8285714285714286)
('Fmeasure of service:', 0.4467265725288832)

Our criteria for passing was 0.6 for precision and recall, both classifier as shown above displays that the accuracy is better than anticipated. We




The most time intensive task of the project was labeling the training data and the test data categories. We manually labeled the over 400 reviews deciding a category for each. This could be a source of error since many of the reviews could be labeled with multiple categories. We also did our best to have equal distributions of each category.

## Review Sentiment Analysis

We also used features from the corpus to do sentiment analysis on the reviews. This is to help up label each review with the reviewers opinions of the business. This is useful for the business to understand their current performance in terms of customer opinion with relation to the categories from the classification section. There are many APIs available to perform sentiment analysis calculations. We choose to continue using the NLTK and the scikit-learn libraries.

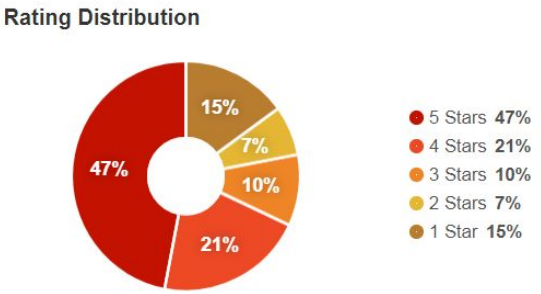
Sentiment analysis was done on a corpus of 1000 reviews. The Yelp dataset contained the star rating of each review, using the user input as the human label. Yelp allows users leave a star rating when they review an establishment, a 1 star rating being the lowest and a 5 star rating being the highest. This allowed us to divided the reviews in 3 classifications for sentiment, a 1-2 star rating is considered a negative review or as used in the software a “neg”. A comment with a 4-5 star rating is considered a positive review, or as tagged “pos”, and a 3 star rating is considered a neutral or “neu” review. In labeling the reviews for sentiment analysis we tried our best to have uniform distributions of our three tags in both the training and test data sets.

The script took each review in the test data bank and labeled it pos, neu or neg using the trained classifier. The data contained the label decided by the user, the review, the classifier label. We used the Naive Bayes Classifier in NLTK and the LinearSVC in scikit-learn libraries respectively to calculate the precision, recall, and Fmeasure for each category. This data was used to calculate each of the accuracy parameters.

('Naive Bayes Classifier accuracy: ', 0.662)
('Precision of pos:', 0.8091397849462365)
('Recall of pos:', 0.8918518518518519)
('Fmeasure of pos:', 0.8484848484848485)
('Precision of neu:', 0.6320754716981132)
('Recall of neu:', 0.4557823129251701)
('Fmeasure of neu:', 0.5296442687747036)
('Precision of neg:', 0.7)
('Recall of neg:', 0.5898876404494382)
('Fmeasure of neg:', 0.6402439024390244)

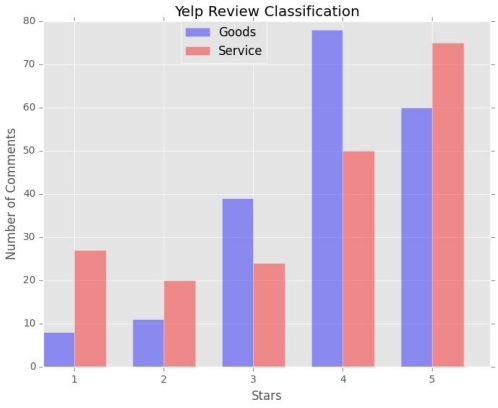
('LinearSVC Classifier accuracy:', 0.664)
('Precision of pos:', 0.8121059268600253)
('Recall of pos:', 0.9540740740740741)
('Fmeasure of pos:', 0.8773841961852861)
('Precision of neu:', 0.5859872611464968)
('Recall of neu:', 0.6258503401360545)
('Fmeasure of neu:', 0.6052631578947368)
('Precision of neg:', 0.6388888888888888)
('Recall of neg:', 0.7752808988764045)
('Fmeasure of neg:', 0.700507614213198)

We decided to trust the user star rating, they are the consumers and the target audience. From the Yelp website, most reviews have 5 star rating with 47% of all reviews, as shown by the image. This could uneven distribution leaves the training set biased in the positive sentiment.

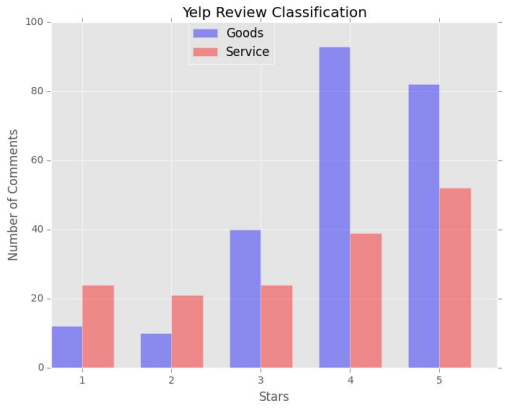



## Data Plot

Finally, having classified each comment as either goods or service we can represent this data in a useful way, using matplotlib, a popular graphing python library. We decided to trust the user star rating. Creating a function to graph both Naive Bayes classifier and LinearSVC classifier in terms of star rating versus total count for each star increment. Since they are two categories we decided  graph each category side by side.




The figure above shows the Naive Bayes classification of each category in terms of the rating it received from the user. Blue is for goods red is for service. With this classifier we can see that the comments with a 1 star rating more users commented on the service received from the establishment. While 4 star rated comments users cared more about the good they received from the establishment.




The figure above shows the LinearSVC classification of each category in terms of the rating it received from the user. Blue is for goods red is for service.With this classifier we can see that the comments with a 1 star rating more users commented on the service received from the establishment. While 4 star rated comments users cared more about the good they received from the establishment. When it comes to the 5 star rated comments there is a difference between the Naive Bayes and LinearSVC. Naive Bayes classified more comments classified as service while LinearSVC has the opposite.

## Discussion and Future Work

The software is effective by being able to show a plot to the business owner showing how many of the reviews input to the classifier are 5 stars due to the goods provided or the service at the business. If the business sees many more 1, 2, or 3 star reviews then they can address where their business is lacking. They are also able to set goals for employees that could lead bonuses or other incentives for meeting goals.
The Naive Bayes and the LinearSVC classifiers have almost equal accuracy to each other.  The LinearSVC has less false negatives when calculating recall than the Naive Bayes giving it higher recall. The precision is about equal in both classifiers, this is very likely because there are reviews where the service and goods classifier is equal. There is probably a third of the reviews that each classifier can decide if the review can be categorized as a good or a

service. Given that all other measures are close to equal we would recommend the the LinearSVC  classifier since it gives us a higher recall than the Naive Bayes.
Future work would definitely include more categories for the classifier. This would allow for a multinomial topic distribution that is capable of expressing more complex content and context for the reviews. We could also further specialize the categories by business sector. For example, we could categorize reviews by ambiance or drinks available for a bar and maybe ensure certain drinks are available certain days. You could also track the peer group of customers and target their specific wants and needs by categorizing  levels of dress.This would allow you to tailor the business to the customers needs. They could The software is intended for the business to scrape or gather their latest reviews and run them through the classifier. The tool will make a shift from a command line interface and manipulation of the source code to a graphical interface, goal would be to make it a web browser tool. That could provide a more detail analysis of reviews, include a graph of the reviews in terms of time to track performance over a period. This will provide a for contextual mining of causal topics with time series supervision. This will better help track the overall performance of the star rating as it relates to certain events influence the the rating of the business. The events can be a personal change, promotional sales, new items, new management, or even a renovation.

## References


[1] ChengXiang Zhai, Sean Massung. Text Data Management and Analysis A Practical Introduction to Information Retrieval and Text Mining, ACM and MC, 2016
[2] Steven Bird, Ewan Klein and Edward Loper. Natural Language Processing with Python. O'Reilly Media Inc.,2014
[3] Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, Iti Mathur. Natural Language Processing: Python and NLTK. Packt Publishing Ltd., 2016
[4] Yelp Dataset Challenge. Retrieved on September 25,2017, from https://www.yelp.com/dataset/challenge
[5] Clark, Scott, Samples for users of the Yelp Academic Dataset ,Latest commit 624534d on Nov 7, 2014, GitHub repository, https://github.com/Yelp/dataset-examples
[6] Yelp Fact sheet, Retrieved on December 12, 2017, from https://www.yelp.com/factsheet
[7] Luca, Michael. "Reviews, Reputation, and Revenue: The Case of Yelp.com." Harvard Business School Working Paper, No. 12-016, September 2011. (Revised March 2016. Revise and resubmit at the American Economic Journal - Applied Economics.)