Complementary Venue Recommendation Model for Yelp

Ryan Wong
Department of Electrical Engineering,
Stanford University

Hyun Sik Kim
Department of Electrical Engineering,
Stanford University

Abstract

Our project attempts to simplify the search process for selecting multiple venues for a single outing using Yelp. We have developed a machine learning model that recommends a complementary venue (such as a café) based on a restaurant searched by a user.

Using a binary classifier, complementary venues were scored (great venue or mediocre / poor venue) based on unigrams and bigrams in review text. Two different models were explored – multinomial Naive Bayes with Laplace smoothing and SVM for a range of kernels.

Overall, Naive Bayes exhibited test error of ~3-4%, comparable to SVM, whilst also being a faster learner and computationally simpler. Transformation of features using term-frequency inverse-document frequency (TF-IDF) along with normalization of review length proved to be key to improving Naive Bayes performance. Among the different SVM kernels, linear was optimal suggesting the text classification problem was linearly separable.

Introduction

Business review websites, like Yelp, are increasingly relied upon by individuals to select an appropriate venue based on past user reviews. However, users often need to select multiple venues for a single outing, such as a neighboring café, dessert house or bar after a dinner at a restaurant, which under the current Yelp interface, must be performed via multiple separate searches.

Our project attempts to simplify this search process and is focused on creating a model (primarily using text analysis) that generates intelligent recommendations for nearby complementary venues (e.g. café) that the user is likely to enjoy based on a primary venue searched.

Effective recommendation systems are valued tools for businesses as they can materially improve the user experience via content personalization.

Related Work

Automatic text classification is becoming increasingly important with the ever growing amount of text information. Naive Bayes and SVM are the two main models that have been commonly applied to prediction of Yelp business star ratings based on review text.

Naive Bayes has been shown to perform well in predicting business ratings despite data not strictly adhering to the conditional independence assumption. Xu, Wu and Wang concluded that binarized Naive Bayes with stop word removal and stemming exhibited the best precision and recall for sentiment analysis amongst perceptron, SVM and nearest neighbor methods [1]. Kibriya et. al showed that standard naive Bayes can be substantially improved by applying a TF-IDF (term frequency – inverse document frequency) transformation to the word features and normalizing the resulting feature vectors for average vector length [2]. Zhang posited that Naive Bayes can still be optimal despite the presence of dependencies if such relationships distribute evenly in classes, or if they cancel each other out [5].

However, the majority of precedent papers have shown that SVM outperforms Naive Bayes in predicting star ratings [3, 4]. Channapragada and Shivaswamy showed that a SVM classifier with adjectives was the optimal model in predicting star ratings [4]. In text classification problems more broadly, SVM is commonly regarded as the method of choice to maximize classification accuracy [2].

Our project aims to apply text classification methodologies in precedent work to the prediction of venue scores for recommendation of complementary venues.

Data

In this project, we used the publicly available Yelp Academic Dataset released under the Yelp Dataset Challenge (round 8), which contains over 2.7 million reviews by 68k users for 86k businesses across Europe and North America. We primarily dealt with three out of the five JSON files summarized in Table 1 below:

Table 1: Yelp Academic Dataset

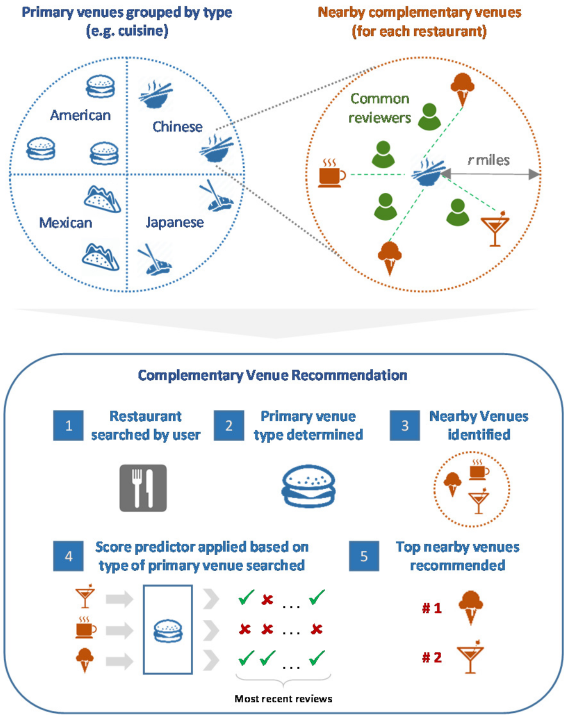
File	Description	Contents
Business	Information on businesses	business ID, name, location, star rating, review count, categories, opening hours and attributes
Review	Review text on businesses	business ID, user ID, star rating, text, date and votes
User	Information on users / reviewers	Type, user ID, name, review count, average stars, votes, social network

Overview of Recommendation Model

Figure 1 below shows our proposed complementary venue recommendation model. Primary venues (i.e. the venue searched by a user) are clustered into *k* groups according to an attribute/s, such as cuisine or price. Complementary venues are nearby venues that the user is likely to enjoy after a visit to the primary venue, such as a bar, café or dessert house.

The system comprises of *k* distinct complementary venue score predictors (i.e. one for each primary venue type). Training of each score predictor is performed based on the set of complementary venues that are within *r* distance of each restaurant within the primary venue type ('nearby venues'). In addition, only reviews of nearby venues by a common reviewer shared with the primary venue are included in the training set.

Figure 1: Proposed recommendation model


Primary venues grouped by type (e.g. cuisine)
Nearby complementary venues (for each restaurant)
American
Chinese
Mexican
Japanese
Common reviewers
r miles
Complementary Venue Recommendation
1
Restaurant searched by user
2
Primary venue type determined
3
Nearby Venues identified
4
Score predictor applied based on type of primary venue searched
5
Top nearby venues recommended
#1
#2
Most recent reviews

When a user clicks on a primary venue, the system first determines the venue type (amongst *k* types). It then shortlists all nearby venues that are within *r* distance of the primary venue. For each nearby venue, the appropriate score predictor (based on the primary venue type) is applied to the most recent *x* reviews. The nearby venue score is the average of these predicted scores. The optimal and hence recommended venues are the ones with the highest average score.

This architecture enables the system to provide recommendations that are both venue and user-centric, whilst maintaining applicability to all users (including casual ones with no review history). Having *k* different predictors enables recommendations to be tailored by primary venue type. User preferences are implicitly accounted for via the training set reliance on common reviewers. This assumes that, for example, a user interested in a French restaurant (primary venue) will likely make similar judgements about a nearby gelato venue (complementary venue) to reviewers common to both venues. Hence, the system does not require any precedent information on the user themselves and can therefore generate recommendations for anyone. Thus avoiding the common cold-start issue.

Text Pre-Processing and Features

For proof of concept, we decided to build a model that provides complementary venue recommendations for a single location, Las Vegas, Nevada. 77 primary venues were selected from the top 200 most-reviewed restaurants for two cuisines – American and Japanese ($k = 2$). Similarly, complementary venues were selected from the top 200 most-reviewed cafes, dessert houses, juice bars & smoothies, and bars that were within $r = 1$ mile distance of the primary venue. From this set of complementary venues, ~5k reviews were randomly selected. Table 2 below summarizes the training set.

Table 2: Training set

Primary venues		Complementary venues		
Type	# of venues	# of venues	# reviews	Avg. review length
American	56	51	5,652	130
Japanese	21	20	5,652	136
Total	**77**	**71**	**11,304**	**133**

Review text was pre-processed prior to token extraction to improve classifier performance:

- Text converted to lower case for standardization
- Punctuation and numbers removed
- Stop words removed to ignore frequent words with no predictive meaning e.g. the, this
- Porter stemming algorithm applied to remove suffixes (e.g. eating -> eat) and hence combine words with the same root in feature vector

We set the feature vector, X, to contain the complementary venue reviews, where $u_i^{(j)}$ denotes, the frequency of the i^{th} word in a unigram vocabulary list, V_1, and $b_i^{(j)}$ denotes, the frequency of the i^{th} word in a bigram vocabulary list, V_2, for review j.

$$X = \left[\begin{array}{cccccccc} \multicolumn{4}{c}{\overbrace{\hspace{8em}}^{\mathit{Unigrams}}} & \multicolumn{4}{c}{\overbrace{\hspace{8em}}^{\mathit{Bigrams}}} \\ u_1^{(1)} & u_2^{(1)} & \cdots & u_{|V_1|}^{(1)} & b_1^{(1)} & b_2^{(1)} & \cdots & b_{|V_2|}^{(1)} \\ u_1^{(2)} & u_2^{(2)} & \cdots & u_{|V_1|}^{(2)} & b_1^{(2)} & b_2^{(2)} & \cdots & b_{|V_2|}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ u_1^{(m)} & u_2^{(m)} & \cdots & u_{|V_1|}^{(m)} & b_1^{(m)} & b_2^{(m)} & \cdots & b_{|V_2|}^{(m)} \end{array}\right]$$

TF-IDF (term frequency – inverse document frequency) transformation was applied to X to enhance classifier performance. Tokens (unigram / bigram) that appear in a review that are infrequent across all reviews are given a higher weighting via the IDF term. Conversely, tokens common to all reviews, and hence likely to have lower predictive power, are given a lower weighting.

$$w_{i,j} = \log(1 + f_{i,j}) \times \log(\frac{N}{df_i})$$

$$f_{i,j} = number\ of\ occurences\ of\ i\ in\ j$$
$$df_i = number\ of\ documents\ containing\ i$$
$$N = the\ total\ number\ of\ documents$$

Finally, each row of counts was normalized to the average review length to correct for reviews of different length.

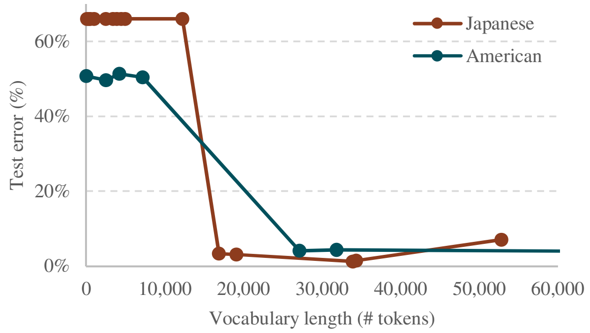
In addition, for SVM a binary feature vector was also investigated, whereby each entry was simply the presence (1) or absence (0) of a token in each review.

The labels vector contained the ratings of each complementary venue review, where $y^{(j)} = 1$ if rating of review $j \geq 4$ stars (great venue) and 0 otherwise (mediocre / poor venue).

The optimal vocabulary list, V (where $|V| = |V_1| + |V_2|$), for each primary venue type, was determined by filtering tokens based on their mutual information score (MI). Figure 2 below shows the test error for varying vocabulary lengths filtered based on MI. The initial drop-off in both series indicates that there is a minimum vocabulary length threshold to achieve a meaningful classifier. The optimal lengths were chosen to be where any further increase in length did not achieve any material increase in test error performance. The optimal vocabulary length corresponded to ~27k tokens for American restaurants and ~17k tokens for Japanese restaurants. Bigrams accounted for more than 80% of tokens across both vocabulary lists.

Figure 2: Test error vs. vocabulary length, |V|

Naive Bayes (Japanese and American restaurants)



Models

We implemented two main types of classifiers – multinomial Naive Bayes with Laplace smoothing (NBM) and SVM as both are well suited and commonly applied to text classification problems.

Under a two-class, multinomial Naive-Bayes model with Laplace smoothing, the parameters, ϕ, are the posterior probabilities of each token, k in vocabulary list, V, for each class. The equation below shows the calculation of the parameters for class $y = 1$. The parameters for $y = 0$ are calculated in a similar fashion.

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} n_i + |V|}$$

Having fit all these parameters, to make a prediction on a new example with features x, the posterior probability is computed for each class (the equation below shows the calculation for class $y = 1$). The predicted class is the one with the higher posterior probability.

$$p(y = 1|x) = \frac{\prod_{i=1}^{n} p(x_i|y = 1)p(y = 1)}{\prod_{i=1}^{n} p(x_i|y = 1)p(y = 1) + \prod_{i=1}^{n} p(x_i|y = 0)p(y = 0)}$$

Under SVM, the parameters, α, are determined based on maximizing the objective function below across m training examples $(x^{(i)}, y^{(i)})$ and for a selected kernel, K. A key advantage of SVMs is that they can learn non-linear classifiers independent of the feature vector dimensionality via an appropriate kernel, such as RBF or polynomial.

$$J_\lambda(\alpha) = \frac{1}{m}\sum_{i=1}^{m}\left[1 - y^{(i)}K^{(i)^T}\alpha\right]_+ + \frac{\lambda}{2}\alpha^T K\alpha$$

Once the parameters are determined, classification is based on computing the hypothesis below, where scores below 0 are denoted class 0 and scores above 0 are denoted class 1.

$$h(x) = \sum_{i=1}^{m} \alpha_i K(x, x^{(i)})$$

The performance of each model was evaluated using the average error rate (misclassification rate) below. $\hat{y}^{(i)}$ denotes the predicted star rating (1 for 4-5 stars; 0 for 1-3 stars) of a complementary venue versus the true star rating, $y^{(i)}$, in the i^{th} review by a common reviewer (ground truth). Since the training set was relatively balanced, average error rate was adopted given its simple intuitive meaning. Test error was determined using k-fold cross validation over the training set with $k = 10$.

$$\text{Error rate} = \frac{1}{m}\sum_{i=1}^{m} 1\{\hat{y}^{(i)} \neq y^{(i)}\}$$

Results and Discussion

Overview

Table 3 and Table 4 below summarize the performance of the two models that were evaluated for each primary venue type – American and Japanese.

For Naive Bayes, classifier performance was evaluated for two different feature vectors – one containing token counts (count), and one with TF-IDF and normalization applied to token counts (TF-IDFN).

For SVM, classifier performance was evaluated for three different kernels (linear, 3rd order polynomial and RBF) and two different feature vectors – one with token counts and TF-IDFN applied (TF-IDFN), and one which simply contained the presence / absence of a token (binary X).

For simplicity, the charts in the following sections are shown for the Japanese restaurants. Similar results were observed for the American restaurants.

Table 3: Summary of model performance (American restaurants)

Model	Training Error	Test Error
Naive Bayes (count)	4.8%	6.0%
Naive Bayes (TF-IDFN)	2.2%	4.1%
SVM – linear kernel (binary X)	2.9%	5.5%
SVM – linear kernel (TF-IDFN)	1.4%	5.1%
SVM – poly kernel (binary X)	0.9%	5.0%
SVM – poly kernel (TF-IDFN)	1.2%	9.5%
SVM – RBF kernel (binary X)	1.8%	6.2%
SVM – RBF kernel (TF-IDFN)	1.0%	9.7%

Table 4: Summary of model performance (Japanese restaurants)

Model	Training Error	Test Error
Naive Bayes (count)	8.6%	9.5%
Naive Bayes (TF-IDFN)	1.8%	3.3%
SVM – linear kernel (binary X)	2.5%	3.8%
SVM – linear kernel (TF-IDFN)	0.5%	2.4%
SVM – poly kernel (binary X)	2.5%	3.9%
SVM – poly kernel (TF-IDFN)	0.2%	3.8%
SVM – RBF kernel (binary X)	1.9%	3.6%
SVM – RBF kernel (TF-IDFN)	0.4%	5.4%

Naive Bayes accuracy comparable to SVM

Naive Bayes with TF-IDFN exhibited test error performance of ~3-4%, which was comparable to SVM of ~2.5-5% (linear kernel). Naive Bayes was robust despite its conditional independence assumption.

Training error was smaller than test error across all models, which suggests a certain level of overfitting (variance error). This was generally more pronounced for SVM. This may have been due to SVM capturing correlations between tokens that did not generalize well, whilst Naive Bayes ignores such relationships under its conditional independence assumption.

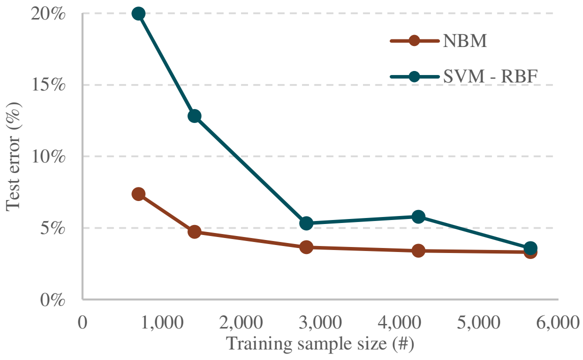
Naive Bayes was also significantly computationally less expensive than SVM given parameters are calculated explicitly using simple probabilistic estimations based on token count versus iterative numerical methods, such as gradient descent in SVM.

Naive Bayes was a fastest learner

Naive Bayes took only ~3k examples, about half of the ~6k examples for SVM to reach asymptotic error performance (refer to Figure 3). Naive Bayes only needs enough data to understand the probabilistic relationships of each token (on an individual basis) with respect to the output variable. Given that correlations between tokens are ignored under the conditional independence assumption, Naive Bayes does not need examples with such interactions and therefore generally requires less data than other algorithms, such as SVM.

Figure 3: Learning curve of NBM vs. SVM

Naive Bayes and SVM (Japanese restaurants)



TF-IDFN was key to Naive Bayes accuracy

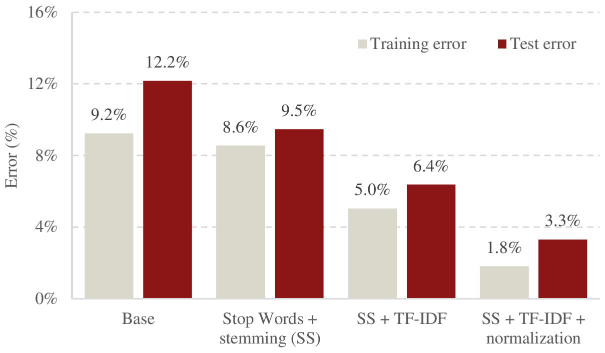
TF-IDFN significantly improved Naive Bayes classifier accuracy by ~6% (refer to Figure 4). In contrast, TF-IDFN had minimal performance improvement for SVM. This importance of TF-IDFN to Naive Bayes is consistent with Kibriya, Ashraf M., et al. [2].

The performance improvement with TF-IDF suggests that the rare words emphasized by the IDF term had greater predictive power over the words common to all reviews.

Normalization corrected for reviews of varying length. Reviews averaged ~130 words in length with some up to 920 words. Normalization ensured that longer reviews were not penalized under Naive Bayes simply due to having a higher token count and hence, higher number of probabilities multiplied during class prediction.

Figure 4: Effect of text pre-processing on performance

Naive Bayes (Japanese restaurants)



Linear kernel optimal

Among the SVM kernels evaluated, the linear kernel with TF-IDFN exhibited the best performance. For Japanese restaurants, linear and polynomial kernels equally provided the best SVM performance of ~5%. This is consistent with text classification problems generally being linearly separable due to the high dimensionality and sparseness of feature vectors. Hence, the added complexity and computational cost of non-linear kernels, such as polynomial and RBF, were ineffective.

Bi-grams improved classifier performance

The addition of bi-grams into the feature vector improved accuracy by ~2% (see Figure 5). The inclusion of bi-grams provided classifiers with the additional predictive power of common two-word phrases.

Figure 5: Unigrams vs. unigrams + bi-grams model

Naive Bayes (Japanese restaurants)

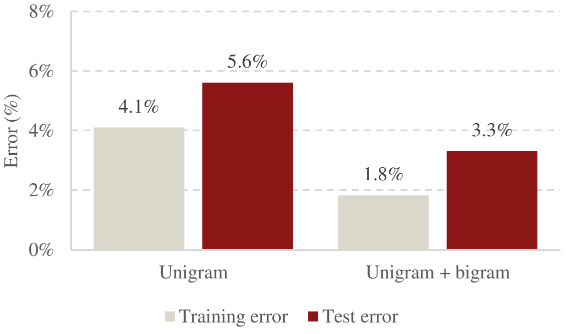
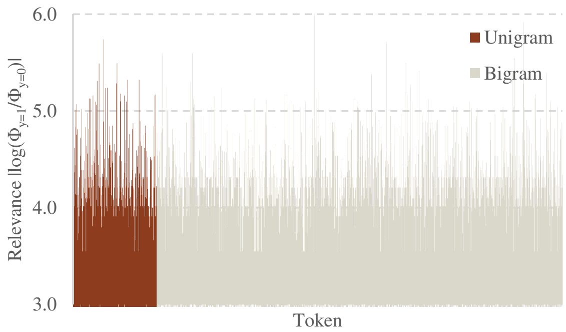


Figure 6 below shows the relevance of each token to the classes based on the absolute value of the log of the ratio of the posterior probabilities. As can be seen, bigrams had comparable relevance in classification to unigrams.

Figure 6: Token relevance

Naive Bayes (Japanese restaurants)

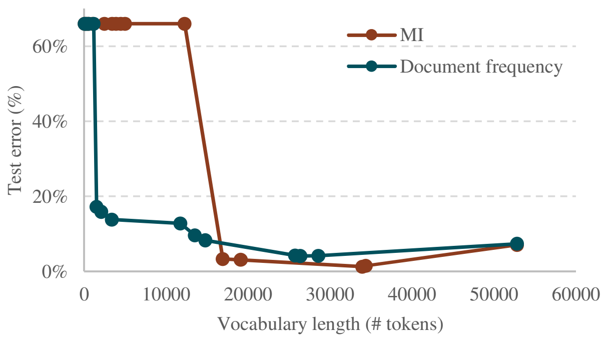


Document frequency was simple and effective for vocabulary selection but MI was better

Figure 7 shows the test error for varying vocabulary length selection using MI and document count. For aggressive vocabulary length reduction (i.e. small vocabulary lists on left hand side), filtering by document count outperformed MI. However, for less aggressive reduction, performance was comparable with MI, with the latter resulting in slightly better performance. This is consistent with [6] in which document count filtering was shown to be simple, effective and broadly comparable to CHI, MI and Information Gain due a strong correlation between scores [5]. The underperformance of MI for aggressive feature selection is likely due to its sensitivity to probability estimation errors from the sample. The effectiveness of document frequency suggests that there is predictive power in common terms (not necessarily just rare terms) in text classification problems.

Figure 7: Feature selection using document count vs. MI

Naive Bayes (Japanese restaurants)



Top positive and negative unigrams and bi-grams

The most relevant positive and negative unigrams and bi-grams were determined by computing the log of the ratio of the posterior probabilities for each token in the vocabulary list of Naive Bayes (refer to Figure 8 and Figure 9 below for Japanese restaurants).

Figure 8: NBM top positive and negative unigrams

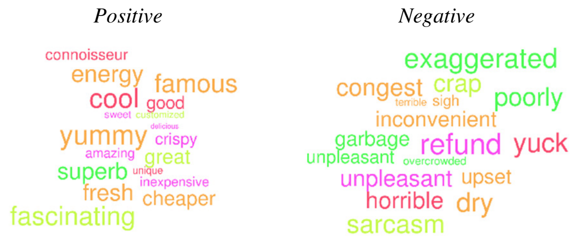
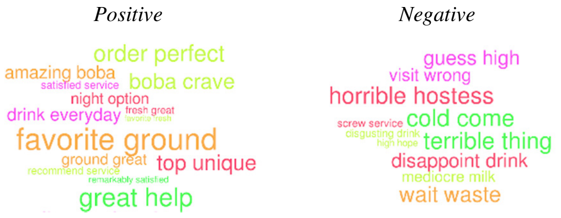


Figure 9: NBM top positive and negative bi-grams



Conclusion and Further Work

We have explored several variations of Naive Bayes and SVM for predicting venue scores based on review text to generate complementary venue recommendations.

Overall, Naive Bayes with TF-IDFN exhibited a test error of ~3-4%, which was comparable to SVM, whilst being simpler and computationally more efficient. Although review text does not strictly adhere to Naive Bayes' conditional independence assumption, this also proved to be one of its advantages. Ignoring correlations between features inherently discourages overfitting and enables Naive Bayes to learn faster than other models. Among the SVM kernels evaluated, the linear kernel was optimal. TF-IDFN and bigrams both materially enhanced Naive Bayes performance.

We plan on performing further work in a number of areas. We would like to develop a second classifier that further distinguishes between 4 and 5 star rated venues within the great venue class to provide finer granularity for venue recommendations. To further improve accuracy, we would also like to explore vector representations of words (word2vec) and n-grams to capture the predictive power of longer phrases and idioms.

References

[1] Xu, Yun, Xinhui Wu, and Qinxia Wang. "Sentiment Analysis of Yelp‘s Ratings Based on Text Reviews." (2015).

[2] Kibriya, Ashraf M., et al. "Multinomial naive bayes for text categorization revisited." Australasian Joint Conference on Artificial Intelligence. Springer Berlin Heidelberg, (2004).

[3] Li, Chen, and Jin Zhang. "Prediction of Yelp Review Star Rating using Sentiment Analysis." (2014).

[4] Channapragada, Sasank, and Ruchika Shivaswamy. "Prediction of rating based on review text of Yelp reviews." (2015).

[5] Zhang, Harry. "The optimality of naive Bayes." AA 1.2 (2004): 3.

[6] Yang, Yiming, and Jan O. Pedersen. "A comparative study on feature selection in text categorization." ICML. Vol. 97. 1997.