CS 229 Final Project Report: Using Yelp Reviews to Improve Businesses

Bonnie Nortz (bnortz) and Stephanie Mallard (smallard)

Abstract

For this project, we investigated running sentiment analysis on topics extracted from review text in order to determine what aspects of businesses are important to customers. The topics for each review were generated from high scoring term frequency-inverse document frequency (TF-IDF) keywords, then used as features for a classification model which would try to predict whether a review was 'extreme' or 'neutral'. After some experimentation, we found that our Support Vector Machine (SVM) model obtained the highest performance and provided topic weights that suggested reviewers care more about quality of the main course and quality of customer service than they care about restaurant gimmicks or perks.

Introduction

While much work has been done on predicting ratings of business reviews based on review word choice, these methods often do not reveal the underlying reasons why customers like or dislike the business they are reviewing. By applying topic modeling to the review text and using the gathered topics as input data, we can perform sentiment analysis that provides more detailed information about the business in question. Since we are using this project for both this class and CS221 Intro to Artificial Intelligence, we decided to use the topic modeling portion of the project for the CS221 project, and focused on the sentiment analysis portion of the project for this class.

Our original project proposal was to perform sentiment analysis on reviews all taken from the same business, in order to determine the business's strengths and weaknesses. Unfortunately, the vast majority of businesses in our dataset had fewer than 400 reviews, which we determined to be insufficient to train and test a reliable classifier. Therefore, we generalized our training and test data to include all reviews from 50 restaurants to gain the necessary data volume. In addition, we simplified our goal output into classifying reviews as 'extreme' or 'neutral' corresponding to star rating, as opposed to directly predicting 1 - 5 stars. This way, weights of features would directly correspond to how strongly a reviewer felt - highly weighted features indicate topics driving 'extreme' reviews and low weighted features indicate customer indifference. One key assumption for this task is that all topics found in reviews that are either very good or very bad are more important to reviewers than all topics found in more neutral reviews.

For our input, we used term frequency-inverse document frequency (TF-IDF) and used the resulting keywords as our topics. We used this method because we found from the CS221 portion of the project that it gave us the best recall within reasonable time constraints. In other words, the topics generated by TF-IDF were most likely to agree with a human annotator. Using the topics of each review as features, we experimented with Naive Bayes (NB) and Support Vector Machine (SVM) models to learn appropriate weights for each topic when classifying documents as 'extreme' or 'neutral'. However, the weight vector of the best classifier is what is truly interesting, because topic weights allows us to draw conclusions about how important certain topics were to reviewers.

Dataset and Features

For this project, we used the data provided by Yelp for their 'Yelp Dataset Challenge' [1]. From this dataset, we chose 16,555 reviews across 50 restaurants, with an average of 331 reviews per

business. We chose 10% of the dataset to use as a test set, giving us 14899 training examples and 1656 test examples. Apart from the review text, the review's star rating (an integer from 1 to 5) was also important to us. Below is the distribution of star ratings across the reviews we considered.

Star Rating	Total Count	Percentage of dataset
1	990	5.98%
2	1311	7.92%
3	2125	12.8%
4	5372	32.4%
5	6757	40.8%

We took further steps to transform our review text into input features for our learning algorithms. First, we did some basic pre-processing of the text, which involved removing extra whitespace, punctuation excluding apostrophes, and a list of stop-words. Then, we ran the text through TF-IDF from the gensim library [3] and kept top-scoring keywords. By choosing a scalable model with the best recall, this means that our final output of weights for each topic will require more analysis to determine which nouns seem like relevant and interesting topics, but it also means that these relevant and interesting topics that a human annotator might pick up on are more likely to be in the list of weighted topics in the first place.

Methods and Results

The first model that we tested to perform our extreme/neutral classification task was Multinomial Naive Bayes, from the sklearn library [2].

Since review star ratings are only integers from 1 to 5, our first challenge was to decide how to define 'extreme' and 'neutral'. Below are the results generated by using both the plain pre-processed text and TF-IDF keywords as input, and by using different definitions of 'extreme' and 'neutral'.

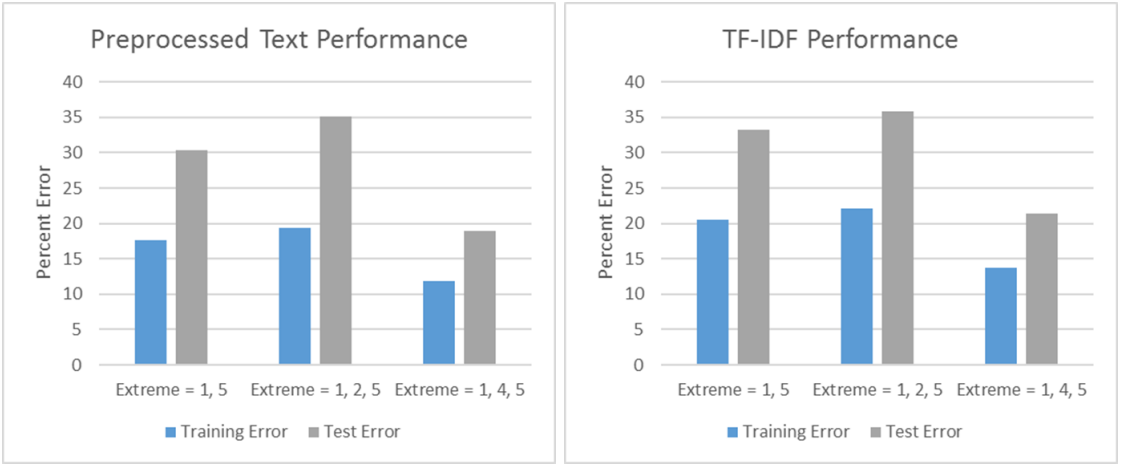

Figure 1: Variations of Label Definitions

As shown, defining 'extreme' to include 1-, 4-, and 5-star reviews gave us the best results.

We noticed that when we were experimenting with a small dataset, the model was more likely to incorrectly predict reviews as neutral rather than extreme, so we attempted to introduce noise into

the dataset to reduce generalization error. We tried switching 5%, 10%, and 20% of reviews with the 'neutral' label to have the 'extreme' label, and while this did make the error more even across 'extreme' and 'neutral' reviews, it did not improve the overall accuracy. Therefore, we decided not to add noise when using the SVM model.

The second model we tested was the SVM. We did similar experiments with definitions of 'extreme', and found similar results as the NB model, so we decided to define 'extreme' as 1, 4, and 5 stars, and 'neutral' as 2 and 3 stars. The next challenge in fitting the SVM was to determine the hyperparameters of number of iterations and step size when running stochastic gradient descent (SGD). To make sure we were not overfitting our model, we plotted learning curves for training and test error and number of iterations, shown below.

Figure 2: Variation of Training Iterations

As shown, the optimal number of iterations is around 65, where the test error is lowest, showing that our model is generalizing well. We also tried a number of different step sizes, and found that a step size of 0.005 works the best.

With 65 iterations and the optimal definition of 'extreme' mentioned above, we generated a final training error of 11.9% and a test error of 18.5%, which is even better than our error for the NB model.

Analysis

One important thing that these results tell us is that because the error was always above 10% for the baseline as well as the TF-IDF input, this is a difficult classification task, and our error rates of 11.9% for training and 18.5% for testing are reasonable, even though other sentiment analysis tasks can get much lower error rates. The learning curves for training and test error for the SVM

also support the fact that we are not simply overfitting the model to the training data, because the test error stays fairly constant after around 40 iterations.

Fortunately, we are not only interested in the error rates of these models, because our main task is to analyze the weights of the topics we ran the models on. Since the SVM performed better, we will only discuss these results. From this model, we generated 33,291 weights ranging from 0.553 to -0.776, with most weights falling between 0.1 and -0.1.

Since our topic modeling involved extracting keywords from documents and not creating more abstract topics, some of the results for the topic weights are rather unhelpful. The table below provides the topics with the highest positive and negative weights. It should be noted that for our SVM, 'extreme' was a (+1) label and 'neutral' was assigned a (-1) label. Therefore, a high positive weight indicates that the topic is important to reviewers, and a very negative weight indicates reviewer indifference.

High Positive	Weights	High Negative	Weights
incredible	0.553	ok	-0.776
outstanding	0.5345	bland	-0.7275
fantastic	0.527	meh	-0.7125
awesome	0.52	average	-0.673
amazing	0.52	mediocre	-0.582
walked	0.449	unfortunately	-0.558

While these words clearly indicate the extremity they are suggested to represent, they do not provide much helpful information about restaurant topics that reviewers find important.

Another difficulty in this type of topic modeling is that similar words will not be grouped into the same topic, so the weights can have sometimes contradictory results. For instance, the word 'employees' has a high positive weight of 0.292, indicating that reviewers felt that the service of the restaurant was an important topic, but then the word 'waiters' had a high negative weight of -0.237, which seems to suggest the opposite. There are a number of examples like this, because the method of topic modeling could not group together words with similar meanings.

However, there were also some very interesting results from the SVM weights. Below is a table with some of the words with the highest positive and negative weights.

High Positive	Weights	High Negative	Weights
delicious	0.4935	overcooked	-0.335
yummy	0.3945	refills	-0.289
authentic	0.31	flavorless	-0.2485
owner	0.2635	option	-0.2465
attitude	0.24	decor	-0.2015
parking	0.2245	boring	-0.1865
atmosphere	0.1765	lackluster	-0.1615
yuck	0.16	beverage	-0.147
poisoning	0.159	toppings	-0.139
disgusting	0.14	appetizers	-0.1325
unappetizing	0.1185	overpriced	-0.1025

The high weights of words such as 'delicious', 'yummy', 'yuck', 'poisoning', 'disgusting', and 'unappetizing' suggest that the reviewer is likely to have a strong opinion if the flavor of the food

is very good or very bad (and perhaps if they got food poisoning). This fact alone might not be very surprising, but the number of flavor words in the top weights does suggest that this is the main point reviewers pick up on, above price and quality of service. Words such as 'overcooked' and 'flavorless' in the negative weights suggest that reviewers will not leave very bad reviews if the only thing against the restaurant is that the food is mediocre. Words such as 'owner' and 'attitude' indicate that reviewers also deeply care about the performance of employees, although not as many of these words appeared in the top weights as did flavor words. Words with the high negative weights such as 'refills', 'beverage', 'toppings', and 'appetizers' seem to suggest that reviewers do not find additional options or gimmicks of restaurants as important as they do the flavor of the main course. This information could be useful to restaurants that are, for instance, considering the trade-off between investing in better ingredients or chefs and just adding more options to the menu.

Conclusion and Future Work

In conclusion, based on our experiments with the Naive Bayes and Support Vector Machine models, we found that we were best able to complete the 'extremity' classification task using 1, 4, and 5 star reviews labeled as 'extreme' and using an SVM model with 65 iterations of stochastic gradient descent and a step size of 0.005. This produced a training error of 13.7% and a test error of 21.4%, which is about the same as the baseline we tested. From this, we analyzed the weights of the topics gathered from running reviews through TF-IDF, and found that quality of the main course and quality of customer service seem to be important topics for reviewers, while restaurant gimmicks do not matter as much.

There are a number of areas where this project could be improved upon in the future. The first of which is to improve the method of topic modeling, which would reduce the dimension of the weight vector gathered from the learning algorithm, and thus the results would not need as much human analysis to determine which topics are relevant and interesting. Also, using more abstract topics rather than keywords would be more likely to divorce individual words from their own sentiment and give us more consistent results. This could be done by improving how we used the models in the CS221 portion of the project, as well as by adding more text pre-processing techniques, such as stemming. It is not clear whether this would make the classification task easier or more difficult, because as shown in the results above, there are a number of nouns that might fall under the same topic that seem to have different 'extremity' values. However, it would probably improve the usefulness of the topic weights.

A second area that could be further investigated is trying other classification models on this 'extremity' classification task. For instance, methods such as a logistic regression or neural networks might produce models better suited to this particular task and might have lower test error than the models we chose.

Finally, future work could involve finding individual businesses with enough reviews about them in order to determine which topics they excelled at and which they failed at. Training a linear regression, weighted linear regression, or similar model to predict exact star ratings from their review text could give weights to topics that would start to answer this question.

References

[1] *More information about the Yelp Dataset Challenge can be found at.* `https://www.yelp.com/dataset_challenge`.

[2] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[3] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.* `http://is.muni.cz/publication/884893/en`. Valletta, Malta: ELRA, May 2010, pp. 45–50.