Sentiment Analysis on Amazon Food Reviews

Jeffrey Chu (j2chu@ucsd.edu)
Ethan Dinh-Luong (edinhluo@ucsd.edu)

I. INTRODUCTION

Sentiment analysis refers to the use of natural language processing and computational techniques in order to understand the subjective nature of human speech and information. Understanding human sentiment language has given many opportunities for companies to understand the user experience of their product and further their innovations in consumer focused ways. However, it has also been one of the biggest challenges in linguistics for many years due to many obstacles such as misrepresented connotations (either by human error or because of complex sentences), differences in meaning between different domains, and understanding of sarcasm. This report aims to highlight one possible way to analyze sentiment in product reviews, specifically Amazon Food Reviews, in order to determine whether the user would recommend the item[1].

The dataset that we used for our model was pulled from Kaggle *Amazon Fine Food Reviews* Dataset with over 500 thousand food reviews on Amazon spanning from October 1999 to October 2012. The dataset contained unique IDs for each product and user, textual reviews, summarized reviews, and product ratings.[2]

To determine whether a user will recommend a product, we analyzed the words in both their textual reviews as well as their summarized reviews to decide if they will give a recommendation. Our model scores each word in a review using term frequency and inverse document frequency (TF-IDF) of the most common words and is applied to a logistic regression model in order to predict recommendations.

The report is organized in the following Sections discussing how the dataset was interpreted and used in the model: Section III discusses the predictive task used and how we evaluated the efficiency of the model, Section IV dives into exploratory data analysis while Section V extracts the features we selected based on our exploratory data analysis, Section VI discusses the procedure of obtaining and running our model with Section VII closing the report by reporting the results of our model compared to baseline models.

II. RELATED WORK

The dataset used in our analysis is an existing dataset used by the Stanford Network Analysis Project on Kaggle, where the data was used as an open ended project to create a machine learning model to predict positive versus negative reviews[2].

With the same dataset, McAuley and Leskovec predict user expertise by first utilizing latent-factor recommender systems on user/item pairs to predict ratings of reviews. With the inclusion of user experience as a measure of temporal information, they are able to model user expertise over time[3].

Karim and Das' sentiment analysis predicted positive versus negative text reviews by scoring the connotative and contextual implications of words in reviews by assigning positive, negative, or a combination of weights to words[4].

Pang and Lee's sentiment analysis to determine positive and negative reviews first used minimum graph cuts to extract subjective portions of reviews then applying a standard classifier on the subjective portion [5].

There are many state-of-the-art sentiment analysis predictors, such as XLNet[6] and ULMFiT[7] that have high accuracies of 96.21% and 95.4%, respectively,[8] on the same dataset.

III. PREDICTIVE TASK

In the dataset, we predicted whether a user will recommend a product based on their textual review and summarized review, where a product rating of 3 and above leads to a recommendation whereas a rating below 3 does not receive a recommendation.

To evaluate the efficiency of our model, we measured both the accuracy and Balanced Error Rate (BER) of the model by identifying the true positive and negative data points as well as the false positive and negative points as the rating scores are imbalanced, where 5-star ratings are predominant in the data compared to all other ratings. To assess the validity of our model, we split half the dataset for training, quarter of the data for validation, and the remaining quarter for the test set. The model was adjusted over time in order to increase the accuracy and decrease the BER of the validation set prior to running the model on the test set. This was done to prevent overfitting of the dataset while also assessing the model on unseen data similar to how it would be used in practice.

There are three relevant baselines that we used to compare our model and aim to improve upon:

Table 1: Relevant Baseline Comparisons

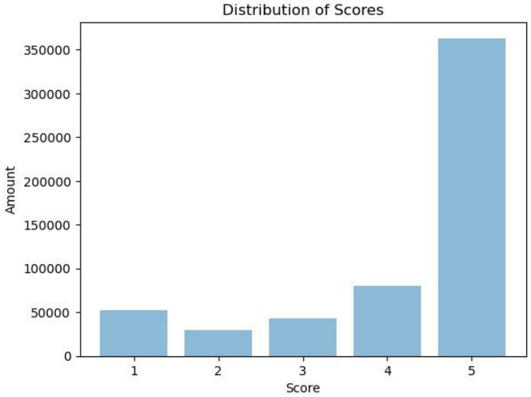
Baseline	Accuracy	BER
I	0.500	0.489
II	0.754	0.498
III	0.831	0.216

I. Random Categorization
II. Probability Categorization based on Proportion of Ratings
III. Only Textual Reviews Used

IV. DATASET ANALYSIS

We first performed EDA in order to understand the shape and distribution of data before choosing certain features. Our first EDA describes the shape of the 'Scores' used in our prediction.

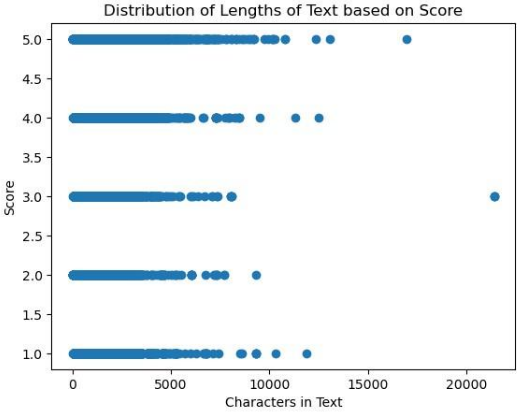
Figure 1: Distribution of Scores



Because we had planned to transform the 'Scores' into binary categories 'User Recommend' for scores greater than or equal to 3 vs 'User did not Recommend' for scores less than 3, EDA performance describes the unbalanced distribution of our output. This led to the decision that our model must recognize this unbalanced data and balance it accordingly.

We then performed EDA on the length of the 'Text' we are using for our feature vector.

Figure 2: Rating vs. Text Length Distribution

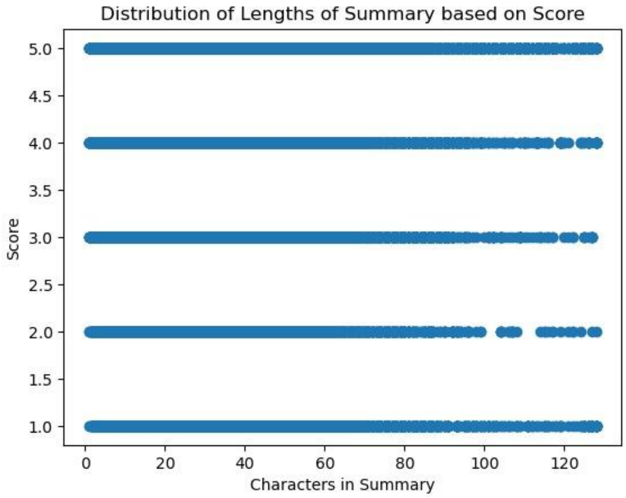


The first occurrence that stood out to us was the large outlier that had over 20000 characters. On closer inspection, this data point had elements of html as well as the 'Text' we were attempting to study. Specifically, there were many accounts of the line break tag with multiple variations of <br>. In theory, the line break is within the set of punctuations and should be treated as such. Thus, as will be explained in our model, this punctuation will be manually removed in our set of features.

The second occurrence that stood was the relatively similar distribution of characters between all the scores. The distribution of characters within the text was similar across different scores (disregarding the outlier with a score of 3), giving us some evidence that it is acceptable to assume that there is no bias within review lengths. That is, the length of the review should not affect whether the user recommended the item (score of 1 or 2) or the user did not recommend the item (score of 3, 4, or 5).

The next set of EDA looks at the length of the 'Summary' that will also be added in our final model. The summary is described as a "brief summary of the review" and the unigrams within the summary were considered just as valid as the unigrams within the 'Text' data.

Figure 3: Score vs. Summary Length Distribution



Examining the EDA tells us that the distribution of summary lengths are fairly uniform across all scores. Similar to the EDA describing the distribution of 'Text' length, this gives us some evidence that it is acceptable to assume that there is no bias within summary lengths and the effect of summary length on the score is negligible.

V. FEATURE SELECTION

Based on our exploratory data analysis, it is acceptable to assume that the length of the text and summary review will have little weight behind the classifier as the lengths are uniform throughout each rating. Because of this, it allows the use of n-grams of the reviews, specifically unigrams, to assign weights of the words to classify the polarity of the review. In theory, there are some contextual and connotative words that could suggest the polarity of a review that would assist in classification.

VI. MODEL

Prior to determining the scores of the most common words, punctuation and stop words were extracted from each review. Punctuation and stopwords serves little weight when determining the recommendation of products, as they do not provide as much information.

To determine the polarity of user reviews, we implemented a standard TF-IDF model on the 1000 most common words, where the score of each word in a review is determined by:

(Eq.1) $term\ freq. \ = \#\ times\ term\ appears\ in\ review$

(Eq.2)

$inv.\ doc.\ freq. = log_{10}(\frac{total\ \#\ of\ reviews\ in\ dataset}{\#\ reviews\ in\ the\ dataset\ containing\ the\ term})$

(Eq.3) $TF - IDF\ score \ = \ term\ freq. \ \times inv.\ doc.\ freq.$

TF-IDF scores were chosen over other metrics such as solely word counts as a weight as TF-IDF adjusts for common, but less significant words by assigning a lower weight, while uncommon, but more significant words receive larger weights.

A 1000 feature matrix was created for each review, where each feature corresponded to the TF-IDF score of the most common words. These matrices were used to train a balanced Logistic Regression model with coefficient 1.0 in order to classify a categorical prediction of 1 for a recommended product and 0 for no recommendation. The 500-thousand review dataset was split into training, validation, and test sets, where the 1000 most common words were determined solely by the training set.

Because we are using the entire dataset, there is a risk of overfitting the dataset, so there are some methods that we applied to reduce the overfitting risk. We shuffled the reviews randomly then split the data accordingly into training, validation, and test sets. Secondly, rather than fixing our model based on the evaluation of the training set, we adjusted our model based on the evaluation of the validation set to avoid overfitting to the training set.

There are many aspects in our model that could optimize our performance with textual reviews.

I. N-grams Greater than 1

A N-grams model could potentially increase our accuracy as groups of words hold more sentiment versus unigrams of words.

i.e.: The unigrams ["pretty"] ["bad"] could hold opposite weights, while the bigram ["pretty bad"] serves a negative weight.

II. **Gradient Descent on Dictionary Size**
Increasing or decreasing our dictionary size from the 1000 most common words could enhance our model, where the use of gradient descent in determining our dictionary size would optimize its performance on the dataset.

III. **Similarity Methods**
Utilizing the user IDs and product IDs in tandem could serve for useful similarity models such as Jaccard and Cosine Similarities, where users rating similar items or items with similar users could enhance our model predictions.

IV. **Weighted Summary Words**
In our model, we counted summary words as if they were a part of a textual review as well rather than considering it as its own dataset. Rather than counting the words, assigning a weight to words in the summary texts could hold more weight as summarized words could hold more sentiment than the raw textual review. The weights of summary words could be optimized using gradient descent to improve our performance.

VII. RESULTS

Using the TF-IDF scores of the words in each review to build a feature vector for a Logistic Regression Model, the following measurements were recorded:

Table 3: Results of the Datasets on the Model

Set	Accuracy	BER
Training	0.860	0.136
Validation	0.859	0.139
Test	0.858	0.138

Compared to the baseline models described in Table 1, the comparisons are as follow:

Table 4: Comparison between Baselines and Model

Model	Accuracy	BER
Baseline I	0.500	0.489
Baseline II	0.754	0.498
Baseline III	0.831	0.216
Model Test Set	0.858	0.138

Based on the comparisons, we can conclude that the inclusion of summarized textual reviews has a small significance in determining the accuracy of the data, as the accuracy only improved by 2%. However, the summaries play a significant role in decreasing the false positive and/or false negative rates, as our BER rate between Baseline III and our model decreased by 8%.

The output of our model produced 1001 parameters, 1000 of which were mapped to a specific word in our features and 1 being the offset parameter, $\Theta = -0.24706393$. For all other parameters, the Θ values can be understood as the “weight” of the word that is has for predicting a positive review. That is, positive Θ values provide more weight that the user recommends the items while negative Θ values provide more weight that the user does not recommend the item. This could be best explained in a table of words with their Θ values:

Table 5: Various Theta Values of Different Words

Word	Θ
great	2.401881027707332
delicious	1.4980411002383252
best	1.4586301820994687
worst	-1.1906665176744025
awful	-0.8922662525520326
terrible	-0.8825457390554151
oil	0.00031091228589943536
skin	-0.0004742250575615124
less	-0.0009584563774962762

Words such as “great”, “delicious”, and “best” have positive meanings in theory and should lead us to believe that the user recommends the item while words such as “worst”, “awful”, and”terrible” all have negative signals that may indicate a negative recommendation. This is reflected on the parameter measures as the positive words also have positive Θ values while negative words have negative Θ values. However, there is some ambiguity in some words such as “oil”, “skin”, and “less”, that don’t really have much value in terms of sentiment and understanding the positivity of the review.

VIII. CONCLUSION

In our report, we analyzed a general sentiment analysis of the words used in food reviews to determine the polarity of each review. A Logistic Regression Model was trained to predict if a review would lead to a recommendation based on a high rating versus no recommendation with a low rating based on these reviews. With our baseline simple text reviews model, we saw a high accuracy using TF-IDF scores to determine the weight of each word across the document, whereas the inclusion of a “summary” reduced the false positive and/or false negative rate significantly. The future of sentiment analysis will allow businesses and public services to better understand people and users to create a comfortable atmosphere for all.

IX. REFERENCES

[1] “Everything There Is to Know about Sentiment Analysis.” *MonkeyLearn*, Retrieved December 08, 2020, from monkeylearn.com/sentiment-analysis/.

[2] Project, S. (2017, May 01). *Amazon Fine Food Reviews*. Retrieved December 06, 2020, from https://www.kaggle.com/snap/amazon-fine-food-reviews/tasks?taskId=797

[3] Mcauley, J. J., & Leskovec, J. (2013). From amateurs to connoisseurs. *Proceedings of the 22nd International Conference on World Wide Web - WWW '13*. doi:10.1145/2488388.2488466

[4] Karim, Mirsa, and Smija Das. “Sentiment Analysis on Textual Reviews.” *IOP Conference Series: Materials Science and Engineering*, vol. 396, 2018, p. 012020., doi:10.1088/1757-899x/396/1/012020.

[5] Pang, Bo, and Lillian Lee. “A Sentimental Education.” *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, 2004, doi:10.3115/1218955.1218990.

[6] Song, Xingchen, et al. “Speech-XLNet: Unsupervised Acoustic Model Pretraining for Self-Attention Networks.” *Interspeech 2020*, 2020, doi:10.21437/interspeech.2020-1511.

[7] Howard, Jeremy, and Sebastian Ruder. “Universal Language Model Fine-Tuning for Text Classification.” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, doi:10.18653/v1/p18-1031.

[8] “Sentiment Analysis.” *NLP*, nlpprogress.com/english/sentiment_analysis.html.