Rating Regional Reviews: Do Regional Differences Impact Sentiment Analysis?

Amani R. Maina-Kilaas
Harvey Mudd College
Claremont, CA 91711
amainakilaas@hmc.edu

Georgia Witchel
Harvey Mudd College
Claremont, CA 91711
gwitchel@hmc.edu

Andy Liu
Harvey Mudd College
Claremont, CA 91711
ajliu@hmc.edu

Harry Weale
Harvey Mudd College
Claremont, CA 91711
hweale@hmc.edu

Abstract

In this paper, we explore how regional differences in the United States can impact sentiment analysis—investigating whether the location of a business causes the language used in its reviews to differ. We tackle this problem through the proxy of comparing rating prediction accuracy. Using the reviews in the Yelp Dataset, we train a rating predictor on the reviews while iteratively holding out different U.S. regions as test sets. The resulting performances show how well each region is modeled by the others, indicating how similar or different they are. Our results show that there is no significant difference and highlight the robustness of sentiment analysis as it pertains to regional differences in the English Language.

1 Introduction

Sentiment analysis has become an incredibly important subtask in the natural language processing field with a wide variety of applications. Sentiment analysis models have been deployed in a variety of real-world contexts, most relevantly in recommender systems. By analyzing a user's sentiment towards specific products or media, a recommender system is better able to recommend new products that the user will feel positively about.

However, one important question surrounding the usage of sentiment analysis in such contexts is the robustness of sentiment analysis models towards regional differences. If residents of different regions in the United States use words in ways that imply different types of sentiments, then this may affect the effectiveness of sentiment analysis in such a context. This question has important implications in the deployment of such systems in real-world applications, as insufficiently robust models may be more accurate in some locales than in others. This might impact the equity of such models, as they may struggle to properly contextualize dialects outside of the regions from which they acquired their training data.

We chose to analyze this question of robustness by considering the proxy of rating prediction accuracy. On the online review website Yelp, users may rate a restaurant from one to five stars based on how much they like it. This is essentially equivalent to a multiclass sentiment analysis task, as the rating given depends specifically on how the user feels about an individual restaurant. We will train rating prediction models on the actual text of a user-written review and the corresponding rating given. To analyze robustness between different geographical contexts, we will train and test our classifiers on different regions of the United States. By seeing if there is a significant difference in performance on different regions, we can analyze how sensitive our models are to regional differences. By doing this analysis, we hope to gain a broader insight towards the effectiveness of sentiment analysis in recommender systems such as those employed by Yelp. We additionally provide the repository for this project[1].

2 Related Work

The analysis of reviews has been extensively studied through various lenses (Finkelstein et al., 2014; Lei et al., 2016; Maalej and Nabil, 2015; Tang et al., 2014; Kouvaris et al., 2018; Luo et al., 2020; Salinca, 2015; Xu et al., 2015).

Maalej and Nabil (2015) used sentiment analysis in combination with metadata, keyword frequences,

[1] https://github.com/amanirmk/rating-regional-reviews

and linguistic rules to classify app store reviews as bug reports, feature requests, or traditional reviews. In their work, they compare results of different preprocessing and classification techniques. With a bag of words (BOW) approach, they found that removing stop words as the only form of preprocessing achieved the highest F1 score—utilizing lemmatization and including metadata improved precision but lowered recall. With respect to methods, they found that Naive Bayes performed better than Decision Tree learning or Maximum Entropy, and that combining multiple binary classifiers was superior to using true multiclass methods.

Lei et al. (2016) proposed a sentiment-based rating prediction method to improve accuracy in recommender systems. Like in our work, they predict the star-level rating based on the text of user reviews. To do their sentiment analysis, they utilized dictionaries of positive and negative words, as well as dictionaries of words indicating the degree or negation of sentiment. Lei et al. also noted that their model performed better on positive rather than negative corpora.

Kouvaris et al. (2018) noted that using raw user star ratings from sites such as Yelp tended to have high levels of noise. While this does not pose an issue in the aggregate, the ambiguity introduced by these star ratings can have a confounding affect on NLP models. As such, they used feature engineering to adjust star ratings to have more parity. By leveraging existing features in user data, they deployed a "super star" rating and demonstrated how it could be utilized in recommendation systems. This research suggests that our task of predicting ratings may be difficult due to the level of noise.

Reviews are also often colloquial, which can introduce its own difficulties. Tang et al. (2014) built a Twitter-specific sentiment lexicon, which they supplemented with definitions from Urban Dictionary. This highlights the difficulty of classification tasks based on colloquial language.

Since reviews are colloquial, it is also more likely for regional differences to be notable and therefore affect the task at hand. Liu et al. (2015) highlighted the topic sensitivity of a simply-trained NLP model, showing that a single model will have vastly different performance based on its training testing split. From this we infer our initial hypothesis: training on a set that uses a different language style, even if subtly, will likely affect the accuracy of the model.

Like us, Xu et al. (2015) predict a venues star review based on its textual rating. They experimented with using Naive Bayes, SVM, and Perceptron, as well as multiple feature selection algorithms. They found that binarized Naive Bayes combined with feature selection with stop words removed and stemming was most effective. This is somewhat surprising, as Frank et al. (2000) showed that Naive Bayes tends to perform more poorly on regression tasks. However, Xu et al. (2015) did note that the Naive Bayes model had very high variance.

3 Methods

3.1 Dataset

We utilize the Yelp Dataset[2] which contains nearly seven million reviews for a hundred fifty thousand businesses centered around eleven metropolitan areas. We use only the reviews from the United States. Each review is given in the form of JSON data, with fields indicating attributes such as business_id, stars, and text.

For convenience, we group the reviews by location and transform the dataset to contain text and star rating pairs. We group states that share borders into clusters so that the geographical locations are more distinct. This results in the following groups: West (AZ, CA, NV), Midwest (IN, MO, IL), South (FL, LA), and East (NJ, PA, DE). We also balance the dataset by these groups, randomly selecting two thousand from each region. In our subset of 8,000 reviews, we find that there are 26,178 unique tokens in the vocabulary, 118 tokens per review on average, and that the average review is rated 3.75 stars with a compound sentiment score of 0.26. Distributions are shown in Figures 1 and 3.

One of the main limitations of the dataset is the number and distribution of areas from which reviews are drawn. As mentioned, the reviews in the United States are centered around specific areas, which do not equally span a wide range of regional areas as would be desired. There is also the possibility of fake reviews influencing the model training or accuracy, but it is still likely that the text of the fake review matches its rating.

3.2 Text Vectorization

There are many possible ways to vectorize the review text. We wish for our results to be independent of how we processed the data, and so we take a few approaches. (1) We use BOW term frequency

[2] https://www.yelp.com/dataset

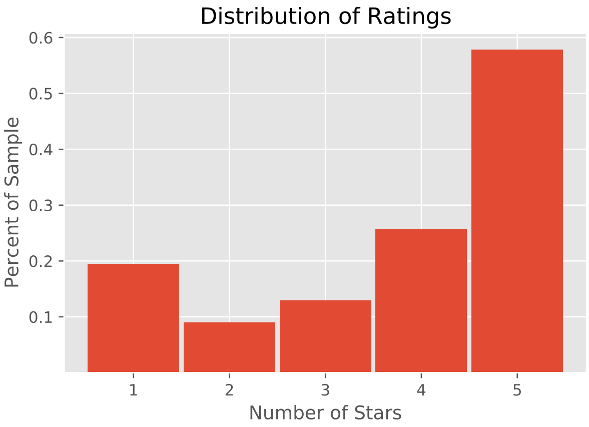

Figure 1: Typical ratings in our 8,000 review sample from the Yelp Dataset.

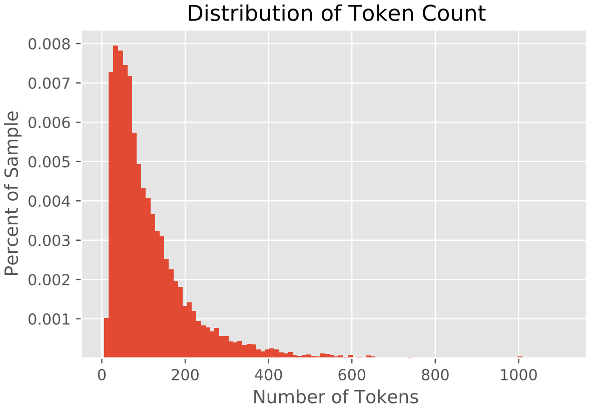

Figure 2: Typical lengths of reviews in our 8,000 review sample from the Yelp Dataset.

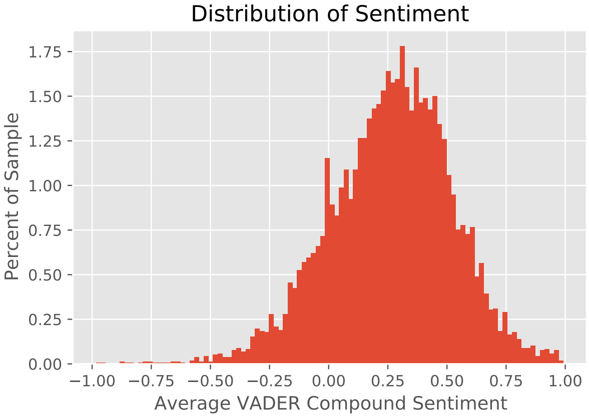

Figure 3: Typical sentiments of reviews in our 8,000 review sample from the Yelp Dataset.

and remove English stop words following Maalej and Nabil (2015), (2) we capture collocational data using bigram frequency, and (3) we supplement BOW and bigrams with the sentiment analysis result from Natural Language Toolkit's (NLTK) pretrained VADER model (Hutto and Gilbert, 2014). This model is optimized for social media text and reports scores for compound, negative, neutral, and positive sentiments. As these scores are by sentence, we average the scores over the sentences in the review. We then concatenate these four values to the existing feature vector. For both BOW and bigrams, we trim our vocabulary to the most frequent one thousand tokens (after removing stop words in the BOW case).

Although we do not consider it here, we note that it may be interesting to utilize Doc2Vec (Le and Mikolov, 2014) as a more semantic text vectorization method.

3.3 Model

We begin our analysis with scikit-learn's Gaussian Naive Bayes Classifier, as the Naive Bayes assumption often works well. However, while Naive Bayes is a common model for classification tasks, it tends to perform worse for regression tasks, even when discretized into categories (Frank et al., 2000). Because of this, we additionally utilize the XGBoost Regressor (Chen and Guestrin, 2016), a scalable tree-based supervised learning algorithm which is widely used for linear and nonlinear regression. For both XGBoost and Gaussian Naive Bayes, we use the default parameters.

3.4 Testing

After combining the metropolitan areas into the four groups as indicated in Section 3.1, we train and evaluate our model in a method analogous to k-fold cross-validation. We train our model on three of the groups and test on the fourth group (a 75/25 split). We repeat this until all groups have been used as a test set. The evaluation involves measuring the average distance from the true rating. This gives us a metric for each group, where worse performance (greater distance) indicates that the group is more dissimilar from the rest. As a baseline, we also perform true, random four-fold cross-validation.

4 Results

We present our results in Table 1. For each model and text vectorization method, we present the aver-

Model	Method	Baseline	West	Midwest	South	East	Overall	Effect (%)
NB	Bigram	1.3207	1.3430	1.2590*	1.3210	1.3115	1.3086	-0.9162
	+ VADER	1.2549	1.2820	1.2115	1.2475	1.2560	1.2492	-0.4542
	BOW	1.1395	1.1155	1.1265	1.1020	1.117	1.1152	-2.1325
	+ VADER	1.0760	1.0615	1.0800	1.0630	1.0685	1.0682	+0.7249
XGB	Bigram	1.0284	1.0639*	1.0238	1.0216	1.0214	1.0327	+0.4181
	+ VADER	0.7179	0.7215	0.7144	0.7142	0.7291	0.7199	+0.2786
	BOW	0.8942	0.9031	0.9049	0.8914	0.8766	0.8940	-0.0224
	+ VADER	0.6885	0.6862	0.6935	0.6966	0.6884	0.6912	+0.3921

Table 1: The average errors in star rating predictions. *Significant at the 95% confidence level.

age distance from the correct rating (error). Baseline is the result of four-fold cross-validation and serves as a reference for comparing the region-specific versions. We also present the results for each region, where for example, "West" means that the West region was used as the test set while the model was trained on Midwest, South, and East. In addition, we show the overall error which combines the region-specific errors. The final column shows the result of $100\% \cdot (\frac{\text{overall}-\text{baseline}}{\text{baseline}})$, which measures the effect of grouping by region as a percentage difference from randomized grouping.

We also run two-sided t-tests for independent samples and find that almost all differences are not statistically significant. Surprisingly, we do find that Naive Bayes and XGBoost with bigram term frequency perform worse at a statistically significant level for the Midwest and West, respectively. We also see that in a few cases, grouping by region performs better than grouping randomly.

5 Discussion

Our results are very clear, we find that the location of the reviews our model is trained on has no clear effect on the model's performance. While two values are statistically significant, there is no observable trend for them, and it is probably by chance. The important column is the overall error, as any difference between baseline and overall is explicitly the result of grouping the reviews regionally instead of randomly. For all values in this column, the two-sided t-test shows that the differences are not statistically significant.

As for observable trends, we find that the addition of the VADER analysis greatly benefits the performance of the model. We also find that using bigrams instead of a simple BOW is worse, likely due to the sparsity of most bigrams. We also note significantly better performance with XGBoost, which in combination with BOW and VADER obtains the best performance.

We now explore hypotheses for why there may be no measurable difference between the grouping methods. One possible reason is that text can be less revealing of regional differences than speech due to lack of accent. Anecdotally, this seems plausible, as one can often guess where someone is from when talking to them, but less commonly when reading their writing. Another reason, is that the one of the main textual clues is slang, which is likely too infrequent to become part of the feature vector. Therefore the model is unable to use these clues in training, preventing overfitting by region. Models that use much larger feature vectors (or more advanced feature selection) may be more likely to use slang as cues. Regional differences may also show up more in the combinations of words, which would be captured by collocational data. Perhaps a more thorough collocational approach would be more affected. It is also possible that our samples are too small to show statistically significant differences, but given the already small effects (at most around 2%), such hypothetical differences would likely be negligible in practice.

Other possible explanations have to do with the Yelp Dataset. People frequently travel, and non-locals may even be more likely to leave a review in the first place. This means that the location of the business may have little correlation with the region of the reviewer, and perhaps we would see different results when grouping by reviewer hometown.

No matter the reason, we find that subtle regional differences in the United States are unlikely to affect the performance of sentiment analysis or recommendation systems. The difference between the training and testing data is not sufficient to cause significant overfitting.

6 Conclusion

In this paper, we considered rating prediction as a proxy to study sentiment analysis systems' sensitivity to differences in regional dialects. We did so by training Naive Bayes and XGBoost model classifiers on Yelp review data while reserving certain location groups before testing on those location groups. We also performed a standard cross-validation on the data as a whole to acquire a baseline without the effect of regional differences. By statistically analyzing our regression results, we concluded that regional differences do not have a significant impact on sentiment analysis outcomes.

If there had been a significant impact, this would have had broader impacts on recommender systems such as those used by Yelp. If such a major component of those systems was sensitive to differences in regional dialect, and performed worse on regions that were underrepresented in their training data, then these effects would also be felt in the recommendations as a whole. Fortunately, it appears our models are robust enough to not be affected by such subtle differences.

In the future, it might be helpful to consider a broader range of sentiment analysis models, such as more neural network-based models that are more common in real-world recommender systems. These extensions would help us better understand the degree to which other sentiment analysis models may be affected by discrepancies in the data. It also might be interesting to consider a similar analysis on all English-language reviews, rather than just those from the United States.

Lastly, it might be worth reconsidering the same data with a different split. Perhaps there is an effect with larger regions grouped, or more specifically with reviewers from different demographics. In general, it is important to investigate the effects of potentially-biased training data and either confirm that our systems are robust, or make dedicated efforts to unbias our data.

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Anthony Finkelstein, Mark Harman, Yue Jia, William Martin, Federica Sarro, and Yuanyuan Zhang. 2014. App store analysis: Mining app stores for relationships between customer, business and technical characteristics. *RN*, 14(10):24.

Eibe Frank, Leonard Trigg, Geoffrey Holmes, and Ian H Witten. 2000. Naive bayes for regression. *Machine Learning*, 41(1):5–25.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Peter Kouvaris, Ekaterina Pirogova, Hari Sanadhya, Albert Asuncion, and Arun Rajagopal. 2018. Text enhanced recommendation system model based on yelp reviews. *SMU Data Science Review*, 1(3):8.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org.

Xiaojiang Lei, Xueming Qian, and Guoshuai Zhao. 2016. Rating prediction based on social sentiment from textual reviews. *IEEE Transactions on Multimedia*, 18(9):1910–1921.

Shenghua Liu, Xueqi Cheng, Fuxin Li, and Fangtao Li. 2015. Tasc:topic-adaptive sentiment classification on dynamic tweets. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1696–1709.

Yi Luo, Liang (Rebecca) Tang, Eojina Kim, and Xi Wang. 2020. Finding the reviews on yelp that actually matter to me: Innovative approach of improving recommender systems. *International Journal of Hospitality Management*, 91:102697.

Walid Maalej and Hadeer Nabil. 2015. Bug report, feature request, or simply praise? on automatically classifying app reviews. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pages 116–125.

Andreea Salinca. 2015. Business reviews classification using sentiment analysis. In *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 247–250. IEEE.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building large-scale Twitter-specific sentiment lexicon : A representation learning approach. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 172–182, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Yun Xu, Xinhui Wu, and Qinxia Wang. 2015. Sentiment analysis of yelp's ratings based on text reviews. In *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, volume 17, pages 117–120.