SENTIMENT ANALYSIS ON TEXT REVIEWS COMPARED TO THEIR RESPECTIVE STAR RATINGS

Len Huang

Holmdel High School

This project was done in the 2018-2019 academic school year in the Honors Advanced Research class offered at Holmdel High School.

Abstract

Python libraries are computer resources with pre-written code that have various uses. The TextBlob NLP library is widely-used in various academic papers regarding sentiment analysis. However, the viability of the library is not always considered. This viability could possibly contribute to error in a sentiment analysis project. To test the accuracy of the library, we drew data from the Yelp Academic dataset, specifically text reviews and star ratings, analyzing them in terms of twelve strata, differing by the number of words each review had. We used TextBlob to get sentiment values of these reviews, then compared library-generated sentiment values on a converted star scale with their respective user-given star ratings. I analyzed and tested a total of 21 samples, two at every word count category. Overall, the p-values from a matched pairs t-test did not give us strong enough evidence to assume that there is a significant mean difference between sentiment star ratings and user-given star ratings. It seemed like reviews with fewer words tend to be closer to the ideal line of y=x, while reviews with more words gravitated towards y=3. In the long run, TextBlob appears to work better with less text than it does more text. While it is not perfect, it is still an effective tool in sentiment analysis projects that analyze passages of text with fewer words.

Introduction

Quantifying textual language is a large area of interest in the field of machine learning. Datasets such as Twitter, Amazon, Yelp, and IMDb have been heavily studied for their large sets of generally succinct and opinionated texts. Gupta, Ravindran, and Shah (2017) use text analysis to build upon the accuracy of review tables on a Yelp data set in an effort to minimize table merging efforts. Mishne and Glance (2009) applied sentiment analysis methods to correlate the qualitative volume of a movie's blog discussion with its financial performance. Go, Huang, and Bhayani (2009) made fundamental progress on such a data set when they employed a Naive Bayes classification model and then used three different selection techniques obtain a high accuracy on classifying sentiment in Twitter messages. In the equation below, c is a class (either "positive" or "negative") and t is the text being analyzed. Thus, the goal is to maximize P(c|t), the probability of the text appearing as either positive or negative. The Naive Bayes method is the main algorithm used in Textblob. Textblob branches off of Python's Natural Language Toolkit (NLTK) library by using a Naive Bayes classifier to produce a polarity score (Explosion AI 2018).

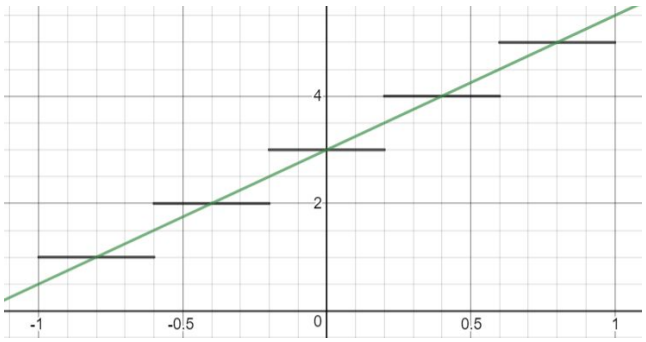
Many studies use such libraries as a starting point to gather sentiment results, then process the data accordingly. Wang and Singh (2018) utilized various deep learning models trained with web-scraped Tweets that were labeled by Textblob polarity scores in an attempt to detect depression through tweets. However, the efficacy of the libraries themselves are not always taken into account. Bari and Saatcioglu noticed this and proposed a novel approach by using the best algorithms in TextBlob, OpinionFinder, and Stanford NLP, and combining into an ensemble framework.

Objective

In this experiment, I will compare the text-analysis abilities of the TextBlob library against text reviews in the Yelp Academic Dataset. The Yelp dataset is a subset of popular review website Yelp.com's businesses, reviews, and user data for use in personal, educational, and academic purposes. When other research is being done, and TextBlob is being used as a tool, researchers can look at this experiment to make an informed decision as to if the library is suited to their work. I hypothesize that the difference between user star ratings and converted star ratings will be significant, and that the average residual will be within a one star difference.

Method

TextBlob's sentiment analysis produces a polarity score from -1 to 1, an indication of how positive or negative the sentiment of a passage of text is. However, Star Ratings on Yelp.com have a score ranging from 1 to 5. I wanted there to be an equal proportion of star ratings as there are sentiment scores, so I decided to divide the polarity into groups of 5, thus giving me a piecewise function incrementing by 0.4. I then attempted to "normalize" the piecewise function into a line of $y = 2.5x+3$. This would produce a score, the "sentiment star rating", reflective of the sentiment score provided.



To see more aspects of how TextBlob works, I decided to test it on twelve different categories: word count. It could be possible that the accuracy of the library varies with the "load" it has to carry, or the amount of words it has to analyze. While my code was parsing through the Yelp Dataset, it simultaneously scanned through each JSON object, counted how many words each review contained, and organized it accordingly.

Then, to see whether or not the data was a result of random sampling, I conducted a matched pairs t-test, with my null hypothesis being that the difference between the sentiment star rating and the original star rating is zero, and my alternative hypothesis being that the difference is not zero. The sample of data from the dataset is inherently random, so we can assume our samples are random as well. Having taken two samples from each word count category, this demonstrates stratified simple random sampling. Each sample contained 30 JSON objects, which is much less than 10% of the population size. By the Central Limit Theorem, we can assume that these samples follow a normal distribution pattern. This was true for all word count categories except for reviews of 1000+ words, where there were only 10 reviews that met this condition. Finally, it is reasonable to assume that each person's opinion and typed review is independent of one another. As such, the conditions of randomness, normality, and independence have been met, allowing us to conduct a matched pairs t-test. I also made scatterplots for each sample, drawing trendlines and displaying coefficients of correlation on each of them. Intercepts were set to zero because it would make no sense for a star of zero stars to return anything besides zero because there are no such reviews.

Findings/Summary

P-Values

	0,50	**50,100**	**100,200**	**200,300**	**300,400**	**400,500**
1	.001083	0.92815	.015936	.300925	.863018	.050433
2	.094072	3.62E-6	5.02E-4	.107187	.155554	9.63E-4
	500,600	**600,700**	**700,800**	**800,900**	**900,1000**	**1000+**
1	.644997	.593695	4.89E-5	.18658	1.25E-4	10 reviews,
2	1.83E-5	.053067	1.37E-4	.565536	2.0237E-5	not enough.

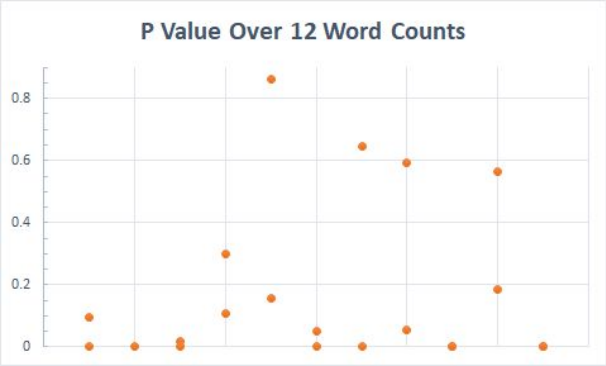
Average Star Ratings

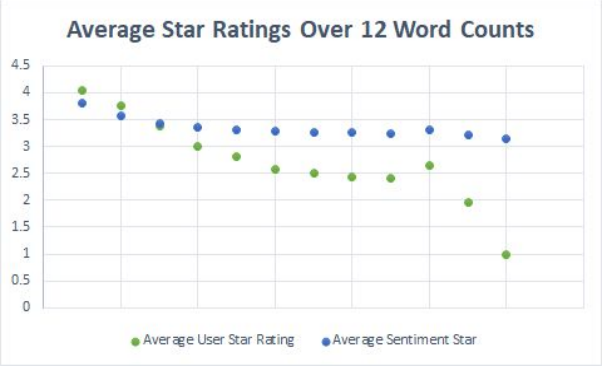
Words	**Average User Star Rating**	**Average Sentiment Star**
0,50	4.047265682	3.816664503
50,100	3.758041486	3.576593895
100,200	3.369483674	3.431969454
200,300	3.007618937	3.348830234
300,400	2.805407105	3.309917281
400,500	2.582793377	3.284020679
500,600	2.510235027	3.270677868
600,700	2.43928036	3.270064464
700,800	2.400584795	3.245907479
800,900	2.648221344	3.311405122
900,1000	1.966019417	3.222841491
1000+	1	3.152461977

Average Residuals

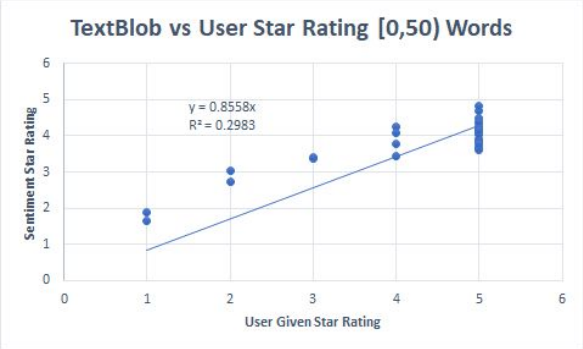
0,50	**50,100**	**100,200**	**200,300**	**300,400**	**400,500**
0.2306	0.18145	-0.06249	-0.34121	-0.50451	-0.70123
500,600	**600,700**	**700,800**	**800,900**	**900,1000**	**1000+**
-0.76044	-0.83078	-0.845323	-0.66318	-1.2568	-2.15246

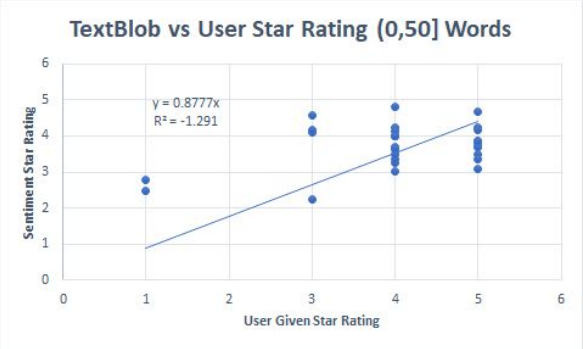
Visualized Data Tables

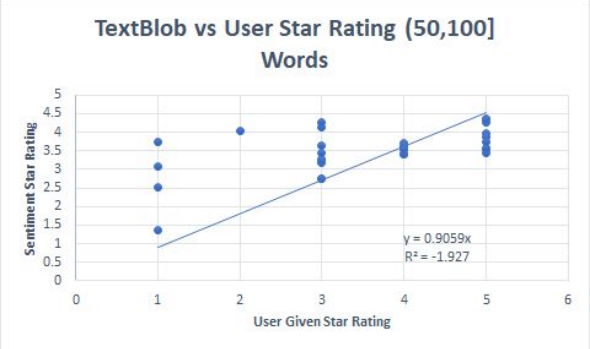


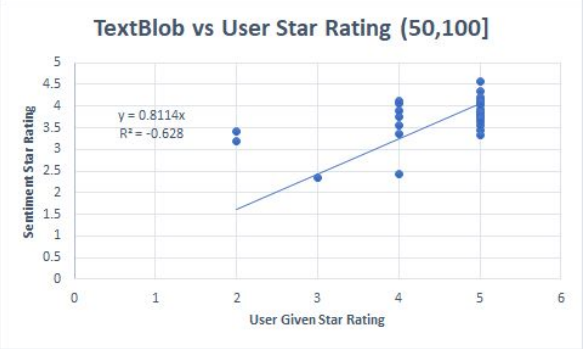


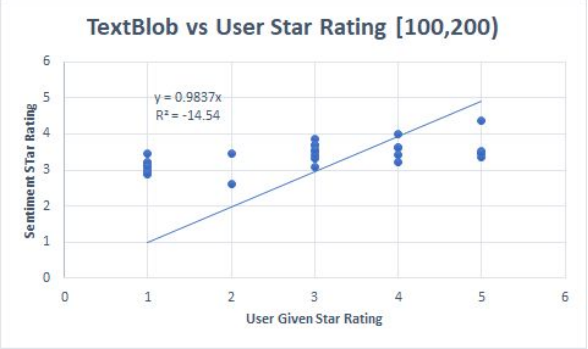
Scatterplots

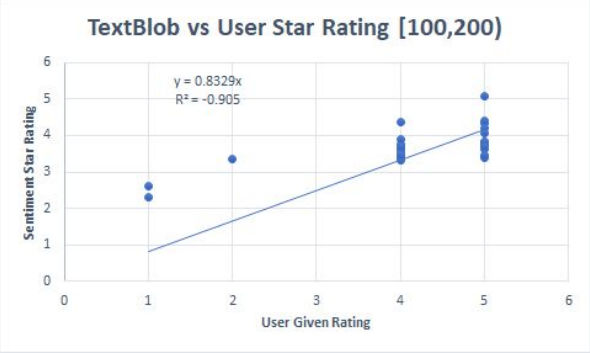


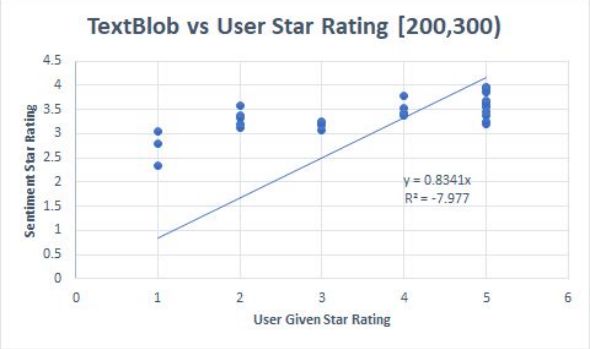


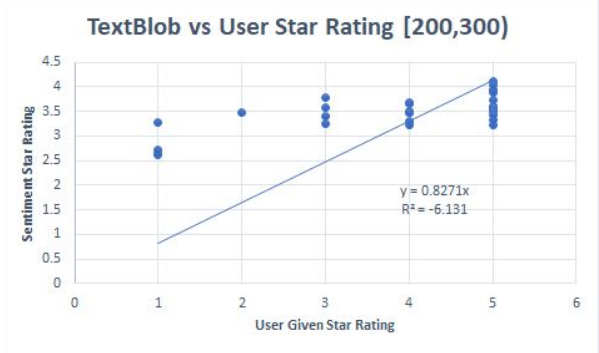


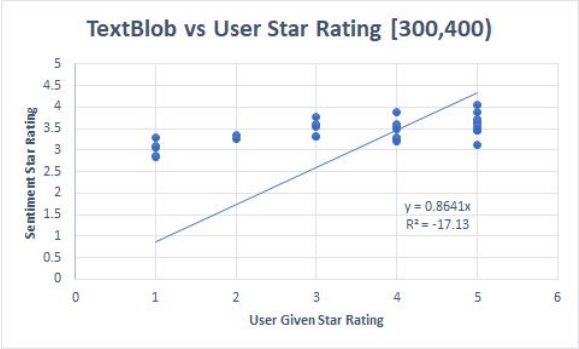


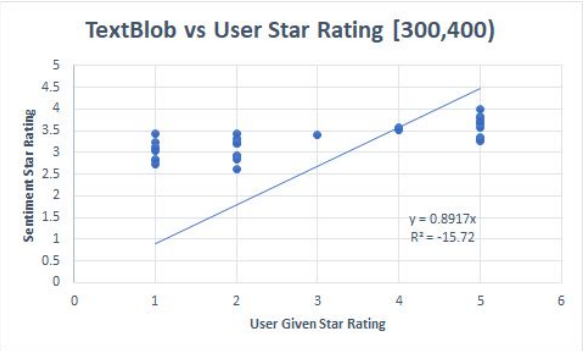


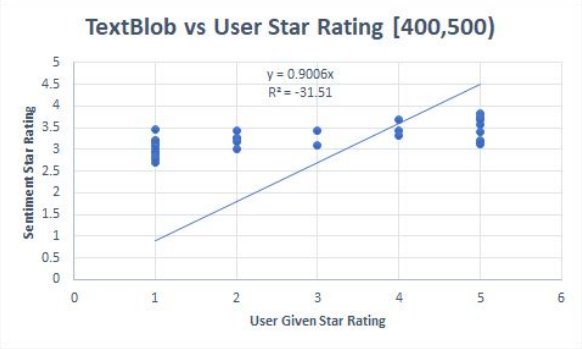


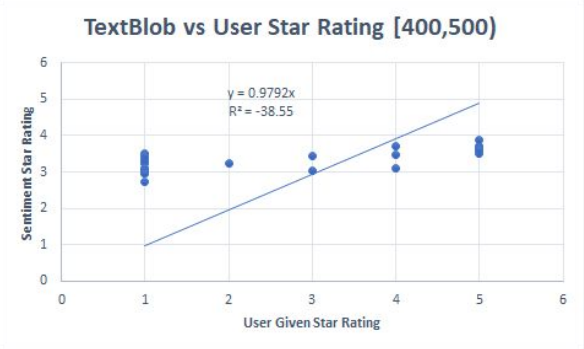


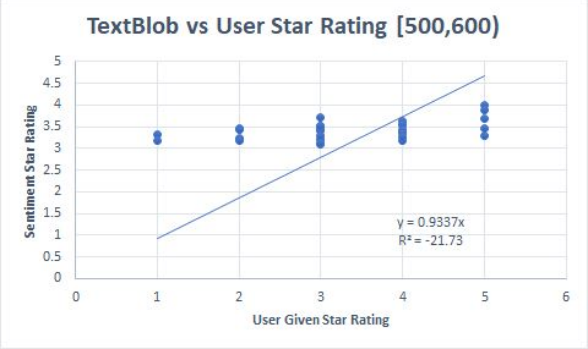


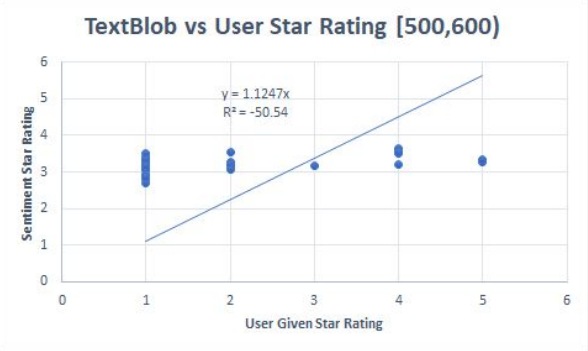

TextBlob vs User Star Rating [300,400)
Sentiment Star Rating
User Given Star Rating
y = 0.8641x
R² = -17.13

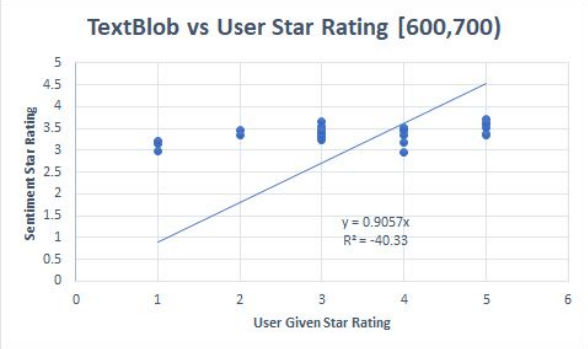

TextBlob vs User Star Rating [300,400)
Sentiment Star Rating
User Given Star Rating
y = 0.8917x
R² = -15.72

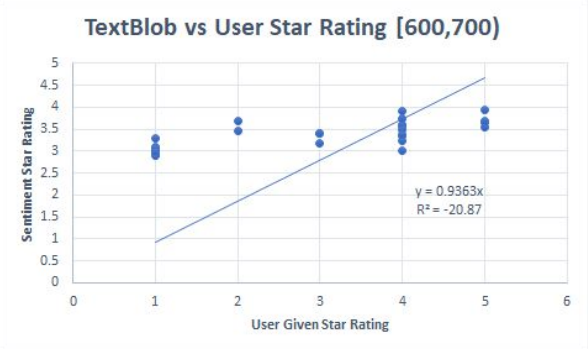

TextBlob vs User Star Rating [400,500)
Sentiment Star Rating
User Given Star Rating
y = 0.9006x
R² = -31.51

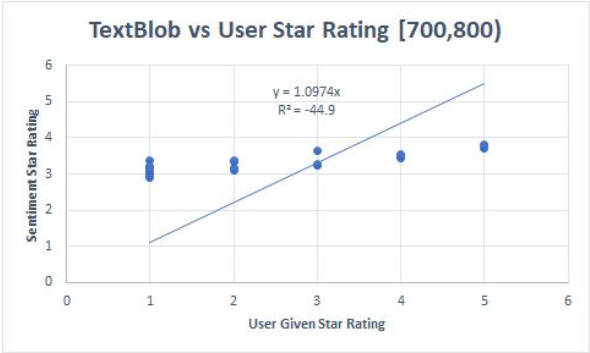

TextBlob vs User Star Rating [400,500)
Sentiment Star Rating
User Given Star Rating
y = 0.9792x
R² = -38.55

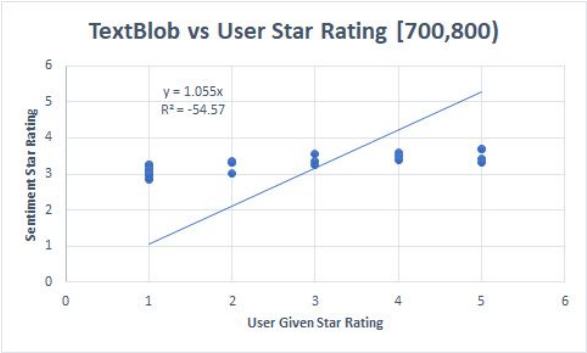

TextBlob vs User Star Rating [500,600)
Sentiment Star Rating
User Given Star Rating
y = 0.9337x
R² = -21.73

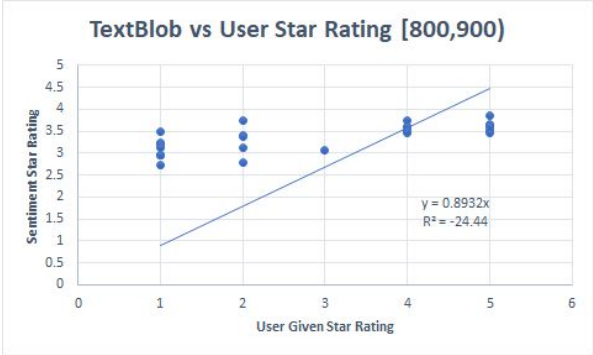

TextBlob vs User Star Rating [500,600)
Sentiment Star Rating
User Given Star Rating
y = 1.1247x
R² = -50.54

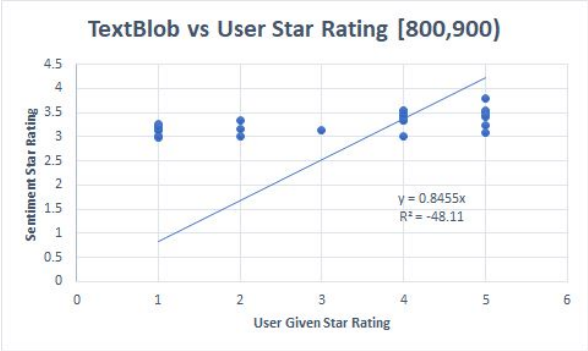

TextBlob vs User Star Rating [600,700)
Sentiment Star Rating
User Given Star Rating
y = 0.9057x
R² = -40.33

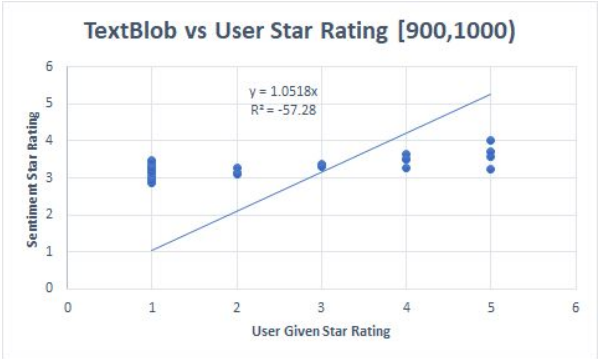

TextBlob vs User Star Rating [600,700)
Sentiment Star Rating
User Given Star Rating
y = 0.9363x
R² = -20.87

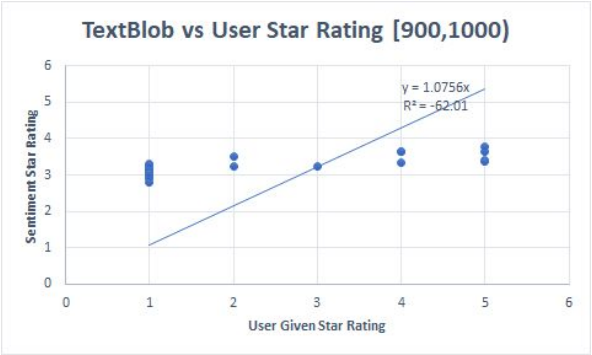

TextBlob vs User Star Rating [700,800)
y = 1.0974x
R² = -44.9
Sentiment Star Rating
User Given Star Rating

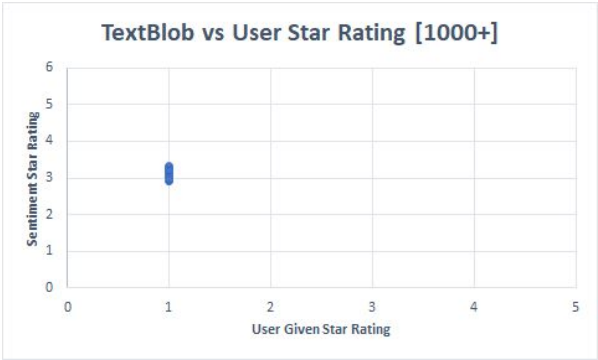

TextBlob vs User Star Rating [700,800)
y = 1.055x
R² = -54.57
Sentiment Star Rating
User Given Star Rating


TextBlob vs User Star Rating [800,900)
y = 0.8932x
R² = -24.44
Sentiment Star Rating
User Given Star Rating


TextBlob vs User Star Rating [800,900)
y = 0.8455x
R² = -48.11
Sentiment Star Rating
User Given Star Rating


TextBlob vs User Star Rating [900,1000)
y = 1.0518x
R² = -57.28
Sentiment Star Rating
User Given Star Rating


TextBlob vs User Star Rating [900,1000)
y = 1.0756x
R² = -62.01
Sentiment Star Rating
User Given Star Rating


TextBlob vs User Star Rating [1000+]
Sentiment Star Rating
User Given Star Rating

Analysis

10/22 of our samples had p values less than the designated alpha level = .05, so there were 10 times where we could reject the null hypothesis that the difference = 0, but 12 times that we failed to reject that null hypothesis. It seemed like reviews with fewer words tend to be closer to the ideal line of y=x, while reviews with more words gravitated towards y = 3.

Discussion

The p-values from a matched pairs t-test did not give us strong enough evidence to assume that there is a significant mean difference between sentiment star ratings and user-given star ratings. This could be because reviews with more words may have more complexity that have various pros and cons balance out to neutral statement, as opposed to a brief review that may just be as simple as a quick, “I love this restaurant” statement.

Conclusion

While Textblob may not be the first thing I resort to when trying to analyze the complex thematic layers of To Kill A Mockingbird, it is still very useful with fewer words. As such, it would make ideal for uses such as analyzing Tweets, as they are often polarized, short, and very opinionated.

Further Study

In the future, this library can be tested with more samples and strata. Instead of word count, a good strata can be what kind of businesses a group is. Perhaps, pizza shops are more accurate than barber shops on average. The same logic of this study can also be applied other libraries, like StanfordCoreNLP, and SpaCy. Stanford CoreNLP is a Java based library with various natural language tools. It’s sentiment analysis is done with a composition model over

trees using deep learning. SpaCy is an NLP library written in Cython, a superset of Python designed to give C-like performance. It's sentiment analysis is done in conjunction with a Keras Long Short-Term Memory (LSTM) classification. LSTM networks are a specific type of Recurrent Neural Network (RNN) that are capable of learning the relationships between elements in an input sequence. Seeing how these various algorithms measure up against the Naive Bayes Classifier of TextBlob can be useful to the scientific community as well.

References

Bari, A., & Saatcioglu, G. (2018, August). Emotion Artificial Intelligence Derived from Ensemble Learning. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (pp. 1763-1770). IEEE.

Go, A., Huang, L., & Bhayani, R. (2009). Twitter sentiment analysis. Entropy, 17, 252.

Explosion AI. (2018, September 15). Text classification with Keras. Retrieved from https://spacy.io/usage/examples

Gupta, O., Ravindran, S., & Shah, V. (2017, October) Text Based Rating Prediction on Yelp Dataset. Retrieved from https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a041.pdf

Loria, S. (2018). textblob Documentation.

Mishne, G., & Glance, N. S. (2006, March). Predicting movie sales from blogger sentiment. In AAAI spring symposium: computational approaches to analyzing weblogs (pp. 155-158).

T Test in Excel: Easy Steps with Video. (2018, September 02). Retrieved from https://www.statisticshowto.datasciencecentral.com/how-to-do-a-t-test-in-excel/

Wang, A., Singh, D. (2018, July, 30) Detecting Depression Through Tweets.

Wang, Y., Yuan, J., & Luo, J. (2015, October). America Tweets China: A fine-grained analysis of the state and individual characteristics regarding attitudes towards China. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 936-943). IEEE.

Working With JSON Data in Python – Real Python. (2018, July 30). Retrieved from https://realpython.com/python-json/

Yelp Open Dataset. (n.d.). Retrieved from https://www.yelp.com/dataset