Sentiment Analysis on Star Reviews Compared to Their Respective Star Ratings

by Len Huang

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. However, for reviews less than 400 words, there was less than 0.5 star difference.

I. 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.

II. 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.

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.

III. Results

P-Values

Word Count	Sample #1	Sample #2
0,50	.001083	.094072
50,100	0.92815	3.62E-6
100,200	.015936	5.02E-4
200,300	.300925	.107187
300,400	.863018	.155554
400,500	.050433	9.63E-4
500,600	.644997	1.83E-5
600,700	.593695	.053067
700,800	4.89E-5	1.37E-4

800,900	.18658	.565536
900,100	1.25E-4	2.0237E-5
1000+	10 reviews	Not enough

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.

Words less than 400 had average differences less than 0.5 stars, and less than 900 had average differences less than 1 star.

IV. 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. 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.

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.

V. References

[1] 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.

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

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

[4] 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

[4] Loria, S. (2018). textblob Documentation.

[5] 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).

[6] 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/

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

[8] 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.

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

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

VI. Acknowledgements

This paper was made possible through the guidance and supervision of Dr. Josephine Blaha at Holmdel High School.