Adam Ruther

ChengXiang Zhai

CS 410 Text Information Systems

5th November 5, 2021

Tech Review: Sentiment Analysis Packages NLTK & John Snow Lab

Sentiment Analysis is a key tool used in Text Mining that can allow us to understand things about an opinion or statement that was made. If you were to search for a Sentiment Analysis Python Package online, you will find many options. However, which is the best one to use is debatable. While there might be a large variety of packages that can perform Sentiment Analysis, I found that there are three model types that are commonly used. These three types include Rule Based, NaiveBayes Classifier, or a Universal Sentence Encoder. NLTK uses Vader which is a Rule Based. TextBlob, SpaCy use a NaiveBayes Classifier and John Snow Labs Twitter Model uses a Universal Sentence Encoder. The two packages that I was most interested in were NLTK and John Snow Labs. NLTK uses a Rule Based Approach while John Snow Labs uses Universal Sentence Encoder which is a machine learning model. I am going to dive into the similarities and contrasts of the two and not only the packages but the model types behind them.

Rule Based approaches were the pioneers of sentiment analysis. Rule Based models, are self-explanatory. There is a set of rules that model pass the input through and provide an output based on the outcome of those rules. The "gold standard" of rule-based sentiment analysis is VADER. VADER has a specific set of rules that is meant to handle social media related data. The base rule for VADER is similar to the Hu-Liu04 model in that it tokenizes each word and checks to see if that word is listed as a positive word or a negative word in a predefined

dictionary. If it is a positive word, we will see a positive constant added to the total score and if it is negative, we will see a negative constant added. Where VADER differs from Hu-Liu04's model is that it has a set of advanced rules to account for social media norms. The rules are attempt to detect tone, sarcasm, slang, and other things that are common in social media that are common in standard text like this paper. These rules include checks like the use of the word **but** (i.e. I like the city of Seattle, **but** it sure rains a lot), an idiom check ("break a leg!"), a special case check (that food was the bomb!), and counting for punctuation (!!!) and emphasis (LET'S GO). Using the SentimentIntensityAnalyzer, the function sums up the overall score and provides a value between 0 and 1 to let the user know how positive or negative the model feels about that statement. While there is a separate python package directly for VADER Sentiment Analysis, VADER is also the sentiment analysis model used by the famous NLP package, NLTK.

In contrast to NLTK, John Snow Labs has sentiment analysis packages. I thought it would be interesting to compare an open sourced package to one that is behind a pay wall. The core model behind John Snow Labs Sentiment Analysis is Universal Sentence Encoder. It was popularized by Google and it its core uses a deep neural network to provide a classification for the sentence based on prior sentences that the model has been trained on. For example, let's say the model is trained on two sentences; the dog played in the park, and "it is very cold outside". Then we run the sentence "it is frigid today" through the model, it is likely that we will pair the sentence with it is frigid today with it is very cold outside. This classification allows us to make certain determinations. For the Universal Sentence Encoder built by John Snows Lab, we use it output an integer value of 1 or 0 to determine whether it is either positive or negative. While this was a very simplified description of the model type there is a lot more to it including different

neural networks that can be used, different encoders, and different transformed. One of the difficult pieces of John Snow Labs model is that we can't exactly see the settings for this model. While it might be frustrating that John Snow Labs Sentiment Analysis Packages are behind a paywall, there are many other models and packages that come with it. The models that are included in the packages that are trained on large specific based datasets. For example, there is a twitter-based version of the John Snow Labs Universal Sentence Encoder model. The specification of these subset models makes up for the overall versatility and broadness of Universal Sentence Encoder based models.

Overall these are two very different model types and the corresponding python packages are vastly different because of it. While, I can't say there is a true winner between the two, I can point out the potential use cases for each of them. NLTK's Sentiment Analysis package is great for twitter. It is specifically built for social media in mind. It is also very easy to explain what is behind this model. If you were to describe this model to a client, you could go into detail exactly what is behind it. This is however the opposite with John Snow Labs Sentiment Analysis algorithm. While it is built specifically for twitter, there are many layers and tasks involved in the neural network that aren't necessarily easily describable. The last contrasting difference is that while the NLTK spits out an exact number, John Snow Labs package outputs a 0 or 1. An exact number allows us to understand a range while the other is a binary result which doesn't allow us to see how exactly positive or negative that output is. To conclude, I believe NLTK is perfect to give a broad explanation on the sentiment while the John Snow Labs Model will be better at classifying exactly if it is positive or negative, even though you might have to pay to utilize this package.

Works Cited

Cer, Daniel, et al. *Universal Sentence Encoder*. 12 Apr. 2014.

Hutto, C., and Eric Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text." *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, 16 May 2014, pp. 216–225, ojs.aaai.org/index.php/ICWSM/article/view/14550/14399.