Professor Zhai
CS 410 Text Information Systems
30 October 2021

LingPipe for Sentiment Analysis of Twitter

In this tech review I will be reviewing the LingPipe toolkit, a toolkit that is used for "processing text using computational linguistics". According to the LingPipe informational website, LingPipe can do many tasks that can extract key information from the news, classify various Twitter results into categories and even develop its own version of autocorrect from queries (http://www.alias-i.com/lingpipe/index.html). Now, for this review I will be concentrating on and researching how we can leverage this toolkit for sentiment analysis. For my final project, I want to implement sentiment analysis on various tweets containing conversations around Covid-19, grouped by region. As I review this specific toolkit, I will be guiding my research around sentiment analysis within the context of my project.

To begin, we can define what sentiment analysis is. According to LingPipe sentiment analysis is when we "classify opinions in text into categories like 'positive' or 'negative' often with an implicit category of 'neutral'" (http://www.alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html). This definition of sentiment analysis closely aligns with my definition as well, but within the context of tweets and Covid-19 might be too vague. For example, when I look at a tweet I would assume the ones containing words like "fake", "scam", "controlling" and "government" would classify as negative, while "science", "frontline" and "second dose" would imply a more positive connotation. This is one of the drawbacks I do see when discussing classification of tweets with respect to the current pandemic, the pre-existing classifications may not consider these new aspects of conversation around the disease. This is what we'll further explore, now that we've deconstructed LingPipe's definition with our problem statement.

The LinPipe toolkit for semantic analysis starts off with what they call a "basic polarity analysis". In the context of tweets, we'd have a dataset that contains tweets and their corresponding locations split into training and test data. The drawback is that in our case, the tweets would not be classified as positive or negative, so this implies that the toolkit relies on pre-classified data to test on, which makes sense because we would not be able to benchmark our code without using these values. I believe that in similar cases where the user has to first construct the dataset itself, such classification can be difficult and inaccurate. This is because our initial dataset itself may have flaws and if we're simply interested in classifying positive versus negative, then this may not be the

best method. A further analysis of the classification portion of this algorithm is necessary.

After analyzing the written analysis of the classification of the dataset, the suspicions of a perfect dataset were correct. The dataset that was used in the example is very much perfect for the algorithm used for basic polarity. In the case of a user who has two parts to this project, one consisting of creating an actual dataset and the other which consists of actual analysis, getting the first part right is crucial to the success of using the basic polarity algorithm provided to us by LingPipe. I believe using the datasets provided in the documentation as a benchmark for what the structure of or resulting text data should look like is the best starting point. After completing this portion, the LingPipe sentiment analysis, specifically the basic polarity, is perfect for such sentiment analysis. In terms of my initial question of whether the polarity would be valid for my specific use case of Covid-19, that still remains unanswered and it's something that is out of question because that is not an easy question to answer without doing extensive coding research. I can conclude that this research will be done as I begin to answer my research questions through actual implementation and I will for sure use this toolkit as a starting point.