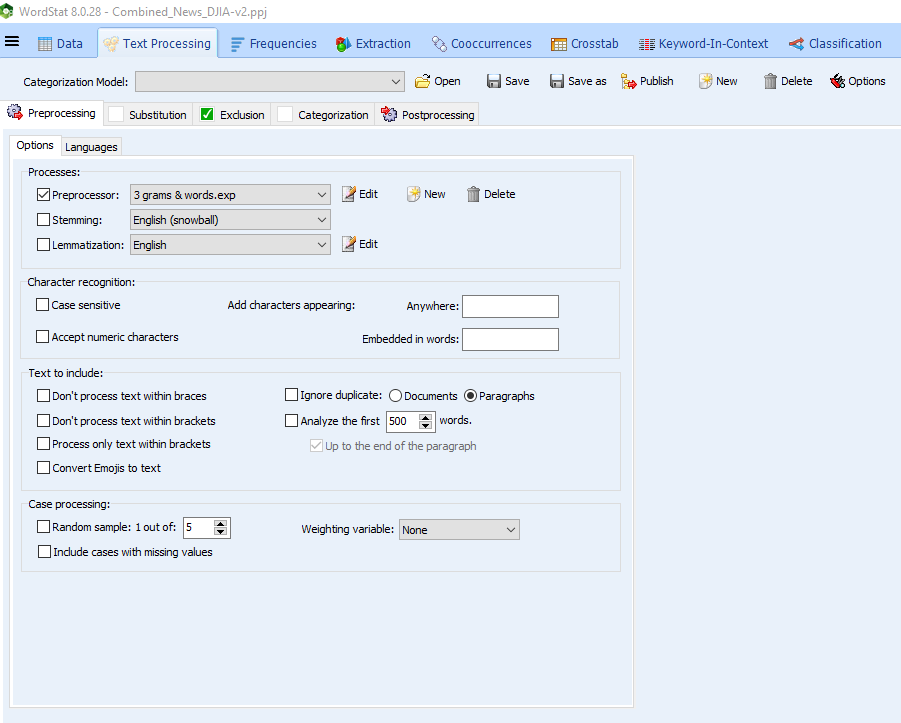
This week’s assignment, we were to use the Kaggle Data Set on News and Stock Prices. I used this opportunity to explore what Word Stat has to offer. For the text processing portion, I found it has several different settings for preprocessing, substitution, exclusion, categorization, and postprocessing.

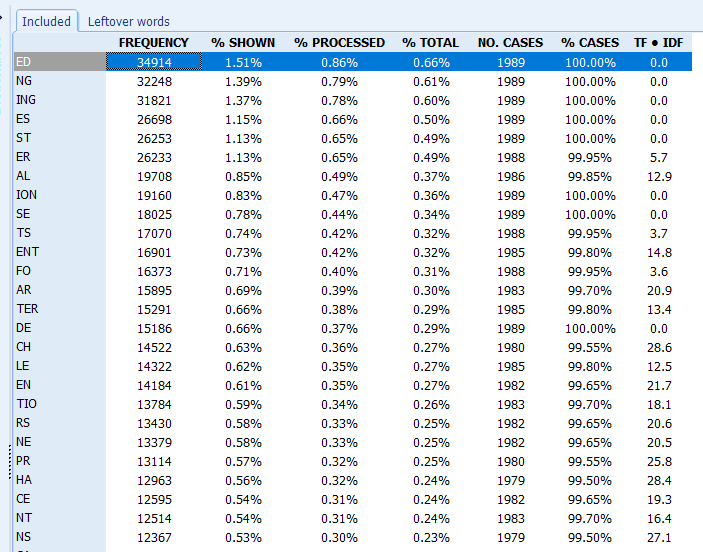
For example, in the preprocessing step, it can calculate 3 grams and word counts, you can choose to perform a stemming and a lemmatization step. If case is important, you can make the processing case sensitive. It even has a feature to convert emojis into text. I found this really interesting especially since social media data is a popular source for text analysis.

You can also choose to import a list to help condense common synonymous words into one so there is less variation on words that are different but have the same semantic meaning. Next it has the exclusion list with details all the stop words to filter from. Word stat gives you the capability to add or remove stop words. The categorization tab allows you to import a dictionary to get the frequency count of phrases. In post processing, you can choose to add words with a frequency or case occurrence high than a particular number or remove items with a frequency/case occurrence less than a particular number. This helps to automate the filtering of rare or common terms. Another option is to remove items that occur over a specified percent of cases or to select a number of items based on a metric such as frequency. These are interesting to me since my final project is a text analysis project. This helps automate the text preprocessing process while in Python, the steps are quite length.

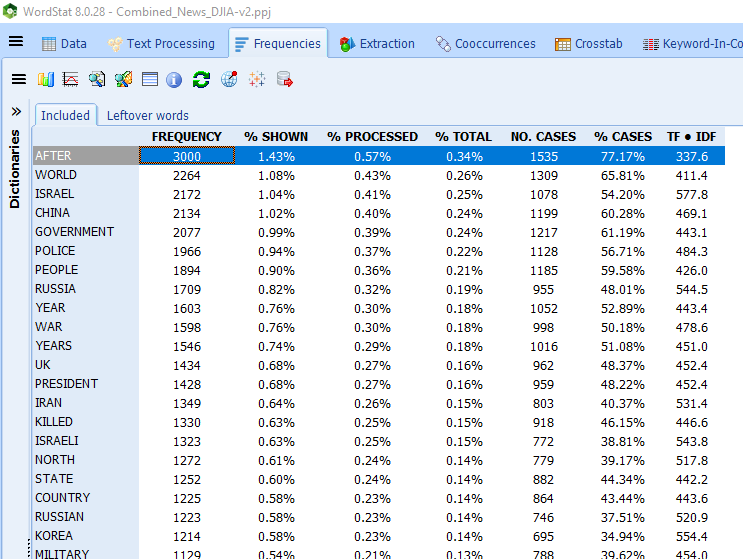


I played around a little bit with the preprocessing settings. When I selected 3 grams and words, I expected the list to become much longer and include 3-word frequencies and one word frequencies but instead, I got a list that looks like popular subfixes. I decided to unselect this processing function that gave me a bag of words frequency count that I did expect.

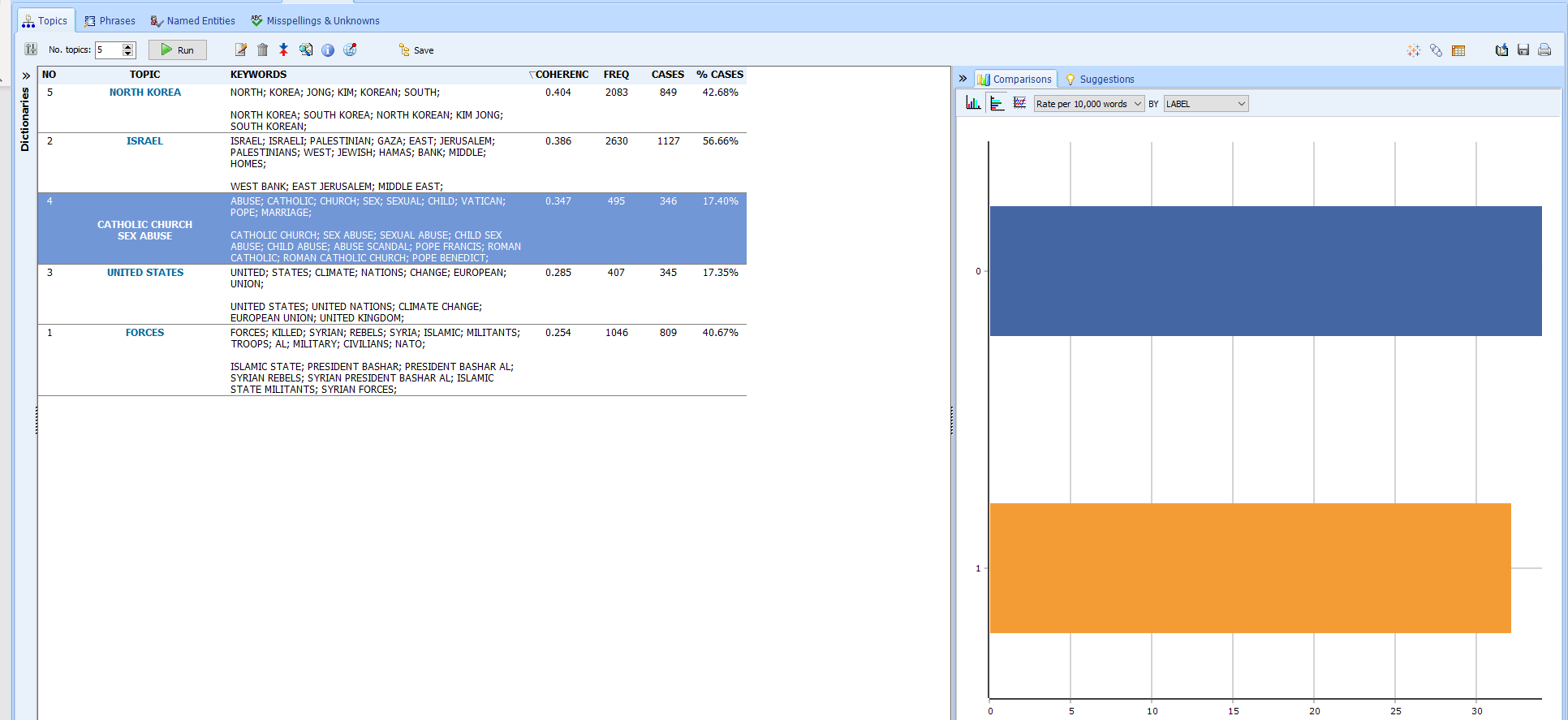
BEFORE:



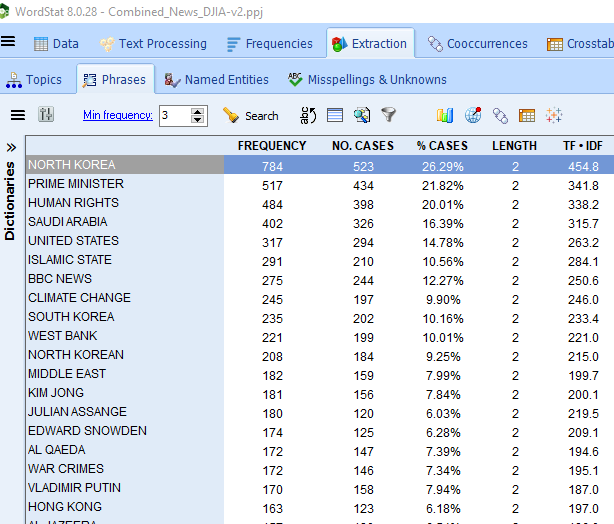
AFTER:



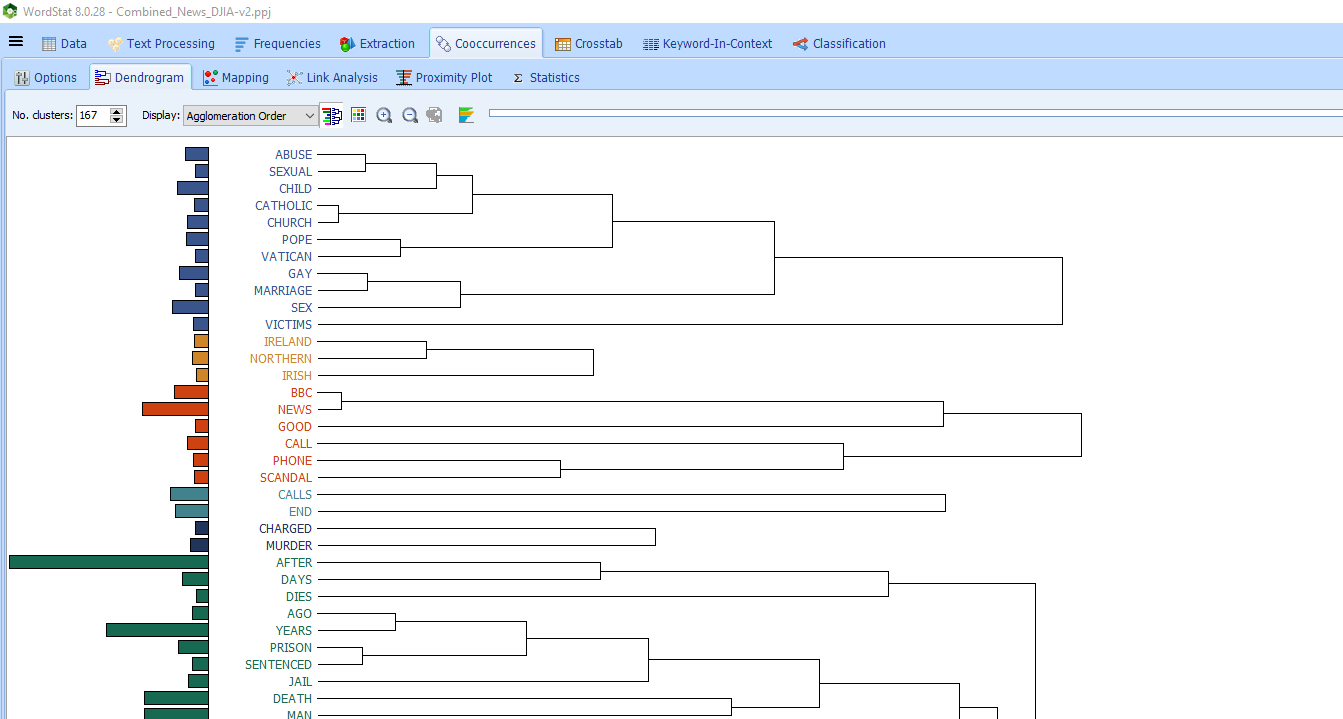
Popular topics included North Korea, Israel, Catholic Church Sex Abuse, the United States, and Forces. Word Stat provided information on how these topics where associated to the label groups on the right side. If it had a more negative impact on the stock price, then the 0 label would be greater than the 1. If there was a more positive effect on stock price, 1 is greater than 0.



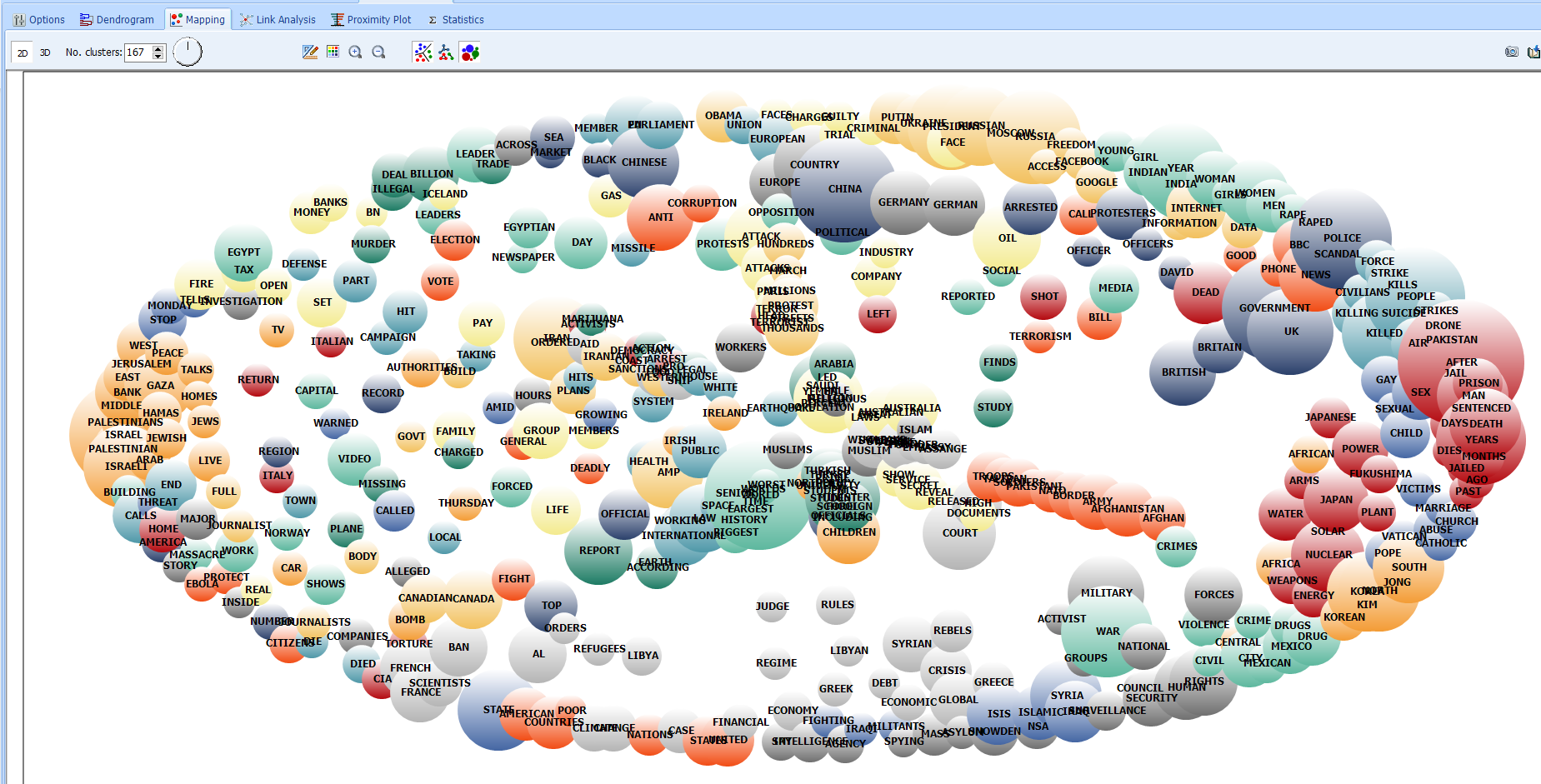
Popular phrases that Dr.Kulich saved as a dictionary during the Webinar can be found under the phrases tab:



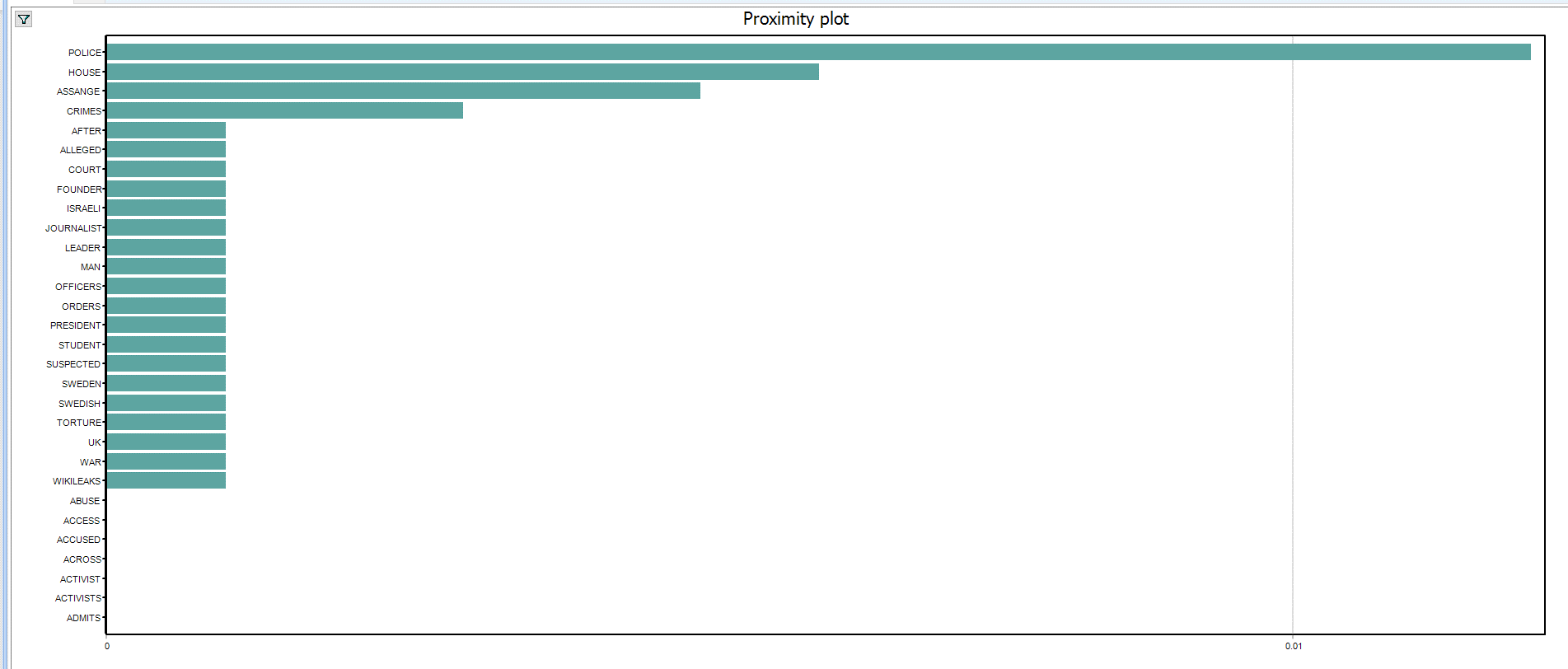
What I found more fun was exploring the coocurrences visualizations that WordStat puts together.



The following is a result of asking WordStat to cluster the words together. I found it interesting that Afghanistan, Afghan, Border, Army, Troops were clustered together. Military, activist, war, groups, national were all not far off. The reason why these words are grouped close together is obvious.

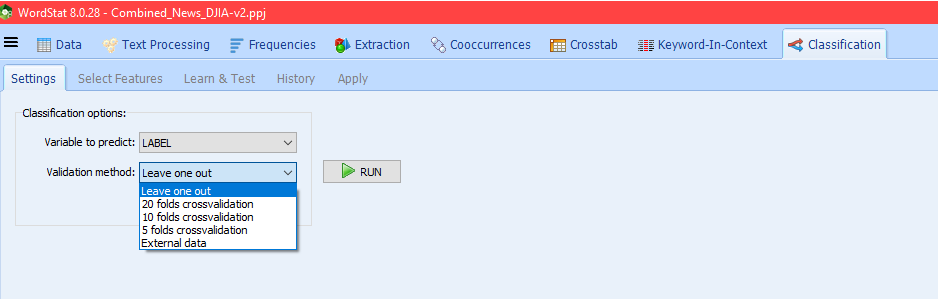


The proximality plot allowed me to choose a word and a visual of other words that occur with that word displayed. For instance, the following is a result of choosing the word “Arrest”:



Police, House, Assange, Crime are among the top of the chart for “Arrest”. I assume the .01 line represents a significance level. If that is the case, “Police” is the only word that is statistically significant to “Arrest”.

Finally, WordStat offers a Classification model that offers the ability to select different models, and validation methods. You can choose to select or leave out feature to include. I decided to the top 100 features with the highest Chi-squared values.



After running a K-Nearest Neighbors model, I received an f1 score of .6607 which represents a balance between precision and recall. The overall precision was .68 and the recall was .6425. With a Naïve Bayes approach, the model performed with lower accuracy with a .59 f1 score, .5981 precision and .5986 recall.