Word Scraping Report

My first step to obtaining the three files was to preprocess the data so it could be handled by a classifier. I began by loading in the data from the articles and the training data and filtering out any stop words, punctuation and incidental phrases like brackets or contraction endings. These all are not relevant to identifying any of the possible subcategories I was looking for so they could be easily tossed. After this I created data frames to train a classifier for each type of response. The features I chose to study were string length, type of string based on the nltk P.O.S. tagger and whether the original string is identical to its stem. Other features using textual context were ruled out because the training data lacks any context to build on so building a model with it wasn’t possible. The P.O.S. tagger and string length were chosen as they are good identifiers of a possible type of string and can be used across all types. The P.O.S. results were transformed into dummy variables for each P.O.S. observed to differentiate them as separate categories for model performance. I chose the stem feature because the desired data consists of mainly formal names and definitions, so most of them are already in stem form and thus could be a good identifier of the desired data. To add to the training data, I chose to use a day’s articles with any of the tagged features removed. Thus, for each category the labeled training data was labeled 1 and the strings from the test article were all labeled 0. Additionally, I standardized the length variable between 0 and 1 in order to make the distance calculations more uniform between variables.

The classification method I chose to use was a Linear Support Vector Machine due to its proficiency with higher dimensional classification problems, which was required due to the large amount of P.O.S. dummy variables. I used a separate SVM for each classification type because each they are looking for very different feature results for each category so combining them in one classifier would likely lower model performance. I also made the SVM’s try to yield a sparse solution due to the large amount of 0’s in dummy variables. I looked at the correct classification rates on the training data for each classifier to judge the accuracy of each model. The accuracy rates for each type were: CEO = 0.7745578344266378, Percentage =0.9284536082474227 , Company =0.9244761106454317 . While the accuracy for CEO’s is a little lower than you’d like to see, this is a bit more expected as it is the most likely of the three categories to be misclassified, especially being mistaken for other names. These trained models were used to select the text files that are attached.