Thomas Cho

IEMS 308 Homework 3

**Features:**

I created the features using Bag of Words model. I removed stop words and preprocessed the text to make it more coherent and together. I used the 1000 most frequent words as my features. I utilize the term frequency inverse document frequency to give those that provide more useful information a higher score than those that don’t.

**Process:**

For each model (Company, CEO, and Percentage), I used ‘bag of words’ to classify the data. Initially, I wanted each instance/occurrence to be a daily text file. However, I noticed that in a single text file, there were several entities, and I had no fair way of classifying each text file. Therefore, I chose to break up the text file into sentences. While reading each text file, I used Regular Expression with a simple logic if a sentence ends with ‘.’, ‘!’,’?’, but that ‘,’ is not between a number or a letter, etc. Then, I appended each sentence as its own row in my DataFrame.

Then, I applied Regular Expression to parse out the entity for the class labels.

For CEO, I took out people’s first names and last names (if given). The big assumption is that their names are right after ‘Chief Executive Officer’ or any of its variations. Then, I eliminated the ‘CEO’ substring using a non-capturing group to obtain my ‘label’ column.

For Company, a company name had to have a abbreviation at the end such as Inc., and the first letter of all words should be Capital. I couldn’t capture companies without such abbreviations.

For percentage, there were two big groups- those that ended with ‘%’ and those that ended with ‘percentage.’ That’s why I made a non-capturing group to capture the letters or numbers or sign (Negative sign) and leave out the percentage out.

Next comes the preprocessing/cleaning the data. For preprocessing…

* I eliminated any characters that are not letters or numbers or % or \. .
* I converted all characters to lower\_case
* I removed stop words that I made (since we couldn’t use any NER tools)
* I removed plural suffixes – s and kept the stem

I removed all sentences without class labels. This dramatically shrunk the data that I would be working with.

**Model:**

For the model, I chose to implement a Naïve Bayes Classifier. It works well with multiclass classifications. It’s efficient in that it is less computationally expensive and quick. Also, Naive Bayes is an example of a high bias - low variance classifier. It’s simple and stable and not prone to over fitting. I used Train/test split of 80/20. There is 2135 records for training, and 534 records for testing for CEO. The classes are CEO names, or company names, or percentage number.

**Performance:**

Accuracy for CEO – 0.45

Accuracy for Company – 0.37

Accuracy for Percentage - .24

**Feedback:**

My method of removing sentences without any entities leaves the model with very little data to work with, and that is why, the accuracy is low.

An alternative solution I tried…

Not many rows had labels or CEO names in those sentences, and therefore a lot of classes had 1-2 instances, and that made the classification models really bad. Therefore, using a big assumption, I filled in labels with the CEO name that was most recently shown. E.g. Let’s say rows 4-7 was blank. Row 3 had “John.” So, I went ahead and filled in “John” for rows 4-7. I did this method because I assumed that there might be a higher probability that those sentences are talking about the last person that was mentioned. It’s not scientific, but I get to use more rows to create a more robust model.

However, the accuracy for CEO was .07. The model is very bad. I suspect that there is no relationship between the word features and the CEO names. I would say there must be a heuristic method. Perhaps, instead of adding the CEO name after every blank, add the name to the next 5 sentences because that represents the average paragraph length or something along that idea might provide better feature selection for the model.