Pranav Kaza

Professor Diego Klabjan

IEMS 308: Data Science and Analytics

5 March 2018

Homework 3: Text Mining and Named Entity Recognition

Part A: Extracting CEO Names

The business text data was initially preprocessed in several steps. First, all punctuation is removed because no features being used necessitate the use of punctuation. In addition, removing punctuation greatly reduces the number of words when tokenizing. After word tokenization based on spaces is done, stop words are removed and bigrams are extracted. Bigrams are used for CEO matching because names are usually 2 words. It is important to note that commonly used preprocessing steps such as normalization, stemming, and lemmatization were avoided to preserve the structure of the entities being searched for, in this case CEO names. For example, the capitalization of the words proved to be important in identifying names, so normalization was avoided. In addition, the feature being used would not benefit from the use of stemming and lemmatization.

To create the training data set, an incomplete training set dictionary of CEO names is used to match from the original text and create positive samples for the classification model. Negative samples are created using 2 methods. First, a dictionary training set of politician names is used because politician names are likely to be mentioned in business articles. In addition, politician names are very similar to CEO names and will therefore make the model more

accurate when training. To avoid class imbalance, additional words are randomly extracted from the text that are not CEO names and used as negative samples as well.

The test data set is created by selecting all capitalized bigrams with the regular expression "[A-Z][A-Za-z]+" that captures all uppercase words and capitalized words. This is done to ensure that the word "CEO" is still in the data set, while all capitalized bigrams that could be CEO names are also captured.

Only one feature was used in the CEO classification model due to the computational difficulties encountered in feature extraction. Feature extraction on the test data set proved to take many hours for a single feature. The feature that was used is:

- Binary feature indicating if the word "CEO" is 30 or fewer characters away from the token. The overall strategy is to find capitalized word bigrams that are near the word "CEO."

Table 1 below shows the summary of the logistic regression model used. The standard error of 0.261 and extremely high z score of 14.338 on the x1 coefficient indicate that the feature was statistically significant. However, the results indicate that the classification model has a heavy positive bias due to overfitting on the single feature that was used; for this reason, many of the extracted names are not accurate CEO names. In a more ideal scenario, more features would be used to create a more accurate model.

```
print(result.summary())
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1853
Model:                          Logit   Df Residuals:                     1852
Method:                           MLE   Df Model:                            0
Date:                Tue, 27 Feb 2018   Pseudo R-squ.:                0.2936
Time:                        11:01:04   Log-Likelihood:               -905.22
converged:                       True   LL-Null:                      -1281.5
                                        LLR p-value:                     nan
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
x1             3.7456      0.261     14.338      0.000       3.234       4.258
==============================================================================
```

Table 1: Logistic Regression Model for CEO Named Entity Recognition

Part B: Extracting Company Names

The methodology for extracting company names was extremely similar to the methodology for extracting CEO names. Because bigrams are once again used as the tokens for study, there is a clear assumption being made in that most company names are 2 words. This is obviously not the case with all companies, but this assumption was made to limit the size of the test data set.

The preprocessing steps used were identical to Part A, and the training and test data sets were created in the same fashion. The feature used for company names was as follows:

- Binary feature indicating if the words “Inc” “Co” “Group” or “Ltd” arre within 10 characters of the token. The strategy is similar in that all capitalized bigrams with these words nearby would be strongly classified as company names.

The logistic regression model used as shown in Table 2 below had an x1 coefficient standard error of 0.201 and a z score of 15.225, indicating strong statistical significance of the

feature. However, the results indicate there is once again a strong positive bias due to overfitting of the model on one feature. In a more ideal scenario with more time and computing power, more features would be used to make the model more accurate.

```
print(result.summary())
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1417
Model:                          Logit   Df Residuals:                     1416
Method:                           MLE   Df Model:                            0
Date:                Wed, 28 Feb 2018   Pseudo R-squ.:                  0.2973
Time:                        08:23:55   Log-Likelihood:                -687.60
converged:                       True   LL-Null:                       -978.44
                                        LLR p-value:                       nan
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
x1             3.0555      0.201     15.225      0.000       2.662       3.449
==============================================================================
```

Table 2: Logistic Regression Model for Company Named Entity Recognition

Part C: Extracting Percentages

To extract percentages, regular expressions were used instead of using supervised learning for named entity recognition. This is because percentages are much easier to identify purely based on surrounded words and symbols than CEO names or company names. It is much more efficient in this case to use regular expressions instead of undergoing the computational and time difficulties of feature extraction and classification for NER. The regular expressions below are used and the lists found and combined into a single list of all extracted percentages:

- "\s[\)\~]?[\+\-]?[0-9]+\.?[0-9]+%[\s\)]?" This is used to identify entries such as "95%" or "-9.6%"
- "\s[\)\~]?[\+\-]?[0-9]+\.?[0-9]+\spercent[\s\)]?" This is used to identify entries such as "-95 percent"

- “\s[\)\~]?[\+\-]?[0-9]+\.?[0-9]+\spercentile\spoints?[\s\)]?” This is used to identify entries such as “+95.3 percentile points”
- “\s[\)\~]?[\+\-]?[0-9]+\.?[0-9]+\spercentage\spoints?[\s\)]?” This is used to identify entries such as “+95.3 percentage points”
- “\s[\)\~]?[A-Za-z]+\spercent[\s\)]?” This is used to identify entries such as “ninety percent”
- “\s[\)\~]?[A-Za-z]+\spercentage\spoints?[\s\)]?” This is used to identify entries such as “ninety percentage points”
- “\s[\)\~]?[A-Za-z]+\spercentile\spoints?[\s\)]?” This is used to identify entries such as “ninety percentile points”

The results indicated 62,991 matches, and all entries are accurately classified percentages. It is important to note that use of a powerful named entity recognition model with many features would likely outperform the use of regular expressions, but given the timeframe of this particular project, the regular expression approach is more reasonable.