**Text Mining Project3**

Hao Zhu

**Executive summary:**

In this case, we have three tasks that extract all company names percentage and CEO names. For each task we have different preprocessing and feature selections, but for the NER we choose the Neural network algorithm and use random over sample method to address the imbalance of samples in my dataset. Also, we analysis the feature we choose by MII and performance of my model by confusion matrix.

**Assumptions:**

* Neglect some special form words that different from other words in one class and count small compared to the entire label dataset. Such as J.K., "2%-2\_%." etc.
* Delete the duplicated names in one article, because too much duplicated samples will influence our results.
* Assume the first letter of words in company names is capitalized.

**Preprocessing:**

* **CEO names**: first we select all the two continuous capitalized words from the txt including the words that has middle abbreviation between those and then use stop word, removing combination that has the stop words. Then, lemmatization. We do not tokenize until feature selection. The Regular Expressions is like picture shown below. (exp1 or exp2)

exp1 = "(?=([A-Z][a-z]+ [A-Z][a-z]+))"

exp2 = "(?=([A-Z][a-z]+ [A-Z]\. [A-Z][a-z]+))"

Also, we find in the prediction example, there is many negative but predicted positive example exist. These examples include large number of Proper nouns, so we decide to exclude them in the preprocess. The Proper nouns includes “North America”, “San Francisco”, “Hong Kong”, “White House”, etc.

* **Companies names**: There are 64 of 4112 company names in label have punctuation. Since if names include punctuation, it would be difficult to find the candidate we ignore the names have punctuation. From the second assumption we can get the Regular Expression is like:

(?:[A-Z][A-Za-z\-]+ ?)+

Next, remove the stop words. However, there are 72 company names have “The”. When removing stop words, do not remove the candidates that have “The”. Like CEO names, we also set some Proper nouns that need to exclude at first.

* **Percentage**: percentage has lots of forms from the label data. We list the most frequent ones.

(-)A% (A can be decimal)

(-)A percent

(-)A percentage

thirty-nine percent…

The Regular Expressions is like words shown below.

For numeric digital: ' (?:((?:-?\d+)(?:\.\d+)?(?:%| percent)(?:age points)?)) '

For text digital: '\w+(?:\-\w+)? (?:percent(?:age points)?)'

**Feature Selection:**

**CEO Names:** The total number of feature kinds is 18. There is one label and one recorded name.

* Whether this name is a CEO name from the label (This is actually a label)
* The candidates of names primary extracted from content (not really a feature, just what we want to extract)

1. Frequency of candidates appear in entire article
2. Frequency of “CEO” appears in entire article. (ignore uppercase)
3. Frequency of “chief” appears in entire article. (ignore uppercase)
4. Frequency of “executive” appears in entire article. (ignore uppercase)
5. Frequency of “officer” appears in entire article. (ignore uppercase)
6. Frequency of “CEO” or “chief”, “executive”, “officer” appears in sentence that name appears. (ignore uppercase)
7. The length of first name (Number of Characters)
8. The length of second name and middle name (if exist) (Number of Characters)
9. The length of one article (Number of Words)
10. The length of all sentences that candidates appear in entire article (Number of Words)
11. Whether the words before candidates are capitalized
12. Whether the words after candidates are capitalized
13. Frequency of “company” in one article
14. Frequency of “company” in the sentences that candidates appear
15. Frequency of “business” in one article
16. Frequency of “business” in the sentences that candidates appear
17. The index of first name
18. Frequency of first name of candidates in entire article

**Companies Names:** The total number of feature kinds is 19. There is one label and one recorded name.

* Whether this name is a company name from the label (This is actually a label)
* The candidates of names primary extracted from content (not really a feature, just what we want to extract)

1. Frequency of candidates appear in entire article
2. Frequency of “company” appear in entire article (ignore uppercase)
3. Frequency of “company” appear in sentence that name appears. (ignore uppercase)
4. Frequency of “CEO” appear in entire article.
5. Frequency of “CEO” or “chief”, “executive”, “officer” appears in sentence that name appears. (ignore uppercase)
6. Frequency of “Financ” appear in entire article (ignore uppercase)
7. Frequency of “Financ” appear in sentence that name appears. (ignore uppercase)
8. Frequency of “Capital” appear in entire article (ignore uppercase)
9. Frequency of “Capital” appear in sentence that name appears. (ignore uppercase)
10. Whether candidates have word “Inc”, “Corp”, “Co”, “Ltd”, “Corporation”, “Capital”, “Company”, “Financial”, “Group” or “Management”.
11. Number of sentences that have candidates’ name in one article
12. Number of Capitalized letter in candidates’ names
13. Length of candidates’ name (Number of Words)
14. Length of candidates’ name (Number of Characters)
15. Length of first word of candidates’ name (Number of Characters)
16. Length of last word of candidates’ name (Number of Characters)
17. Length of entire article
18. Length of all sentences that candidates appear in entire article
19. The index of candidates in sentence.

**Percentage:** For the percentage, we do not need to set features because there are few confusions in extracting. We can assure that digital plus % or percentage or percent is a percentage. So, we only need to use RE to select.

**NER Classifier train:**

First, for the NER we choose the SVM algorithm and use package of sk-learn to train my model. However, it takes too much time, which means SVM is not appropriate for the large dataset. I change the algorithm into Neural network models. Set 70% of data as training set, 30% data as testing set.

The sample distribution is imbalance (take CEO names for example, there are 9738 positive samples but 290526 negative samples in data set). In this case, we use method of random over sample to balance the negative samples and positive samples.

**Feature Selection Analysis:**

In this project, we use Mutual information and maximal information coefficient to measure the importance of features.

**CEO Names:**

We get the coefficient of 18 features:

[0.00264241 0.00041983 0.00014946 0.00018197 0.00015962 0.01267916

0.00667041 0.0047614 0.00211451 0.00693079 0.0034555 0.00504393

0.00033207 0.00180235 0.00027003 0.00032589 0.00099453 0.00160503]

Frequency of “CEO” or “chief”, “executive”, “officer” appears in sentence that name appears. (ignore uppercase) seems like the most relative feature in this model.

**Companies Names:**

We get the coefficient of 19 features:

[0.00579616 0.00028424 0.00328903 0.0001841 0.00226616 0.00029142

0.00088833 0.00011444 0.00098808 0.00808977 0.00552397 0.00174471

0.00106143 0.00404608 0.00399041 0.00292125 0.00139021 0.01099465

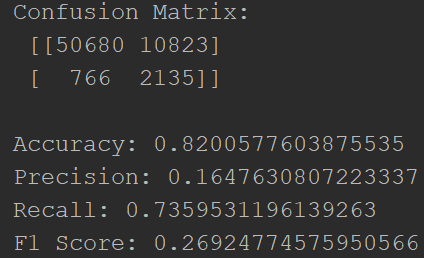
0.00471215]

Length of all sentences that candidates appear in entire article seems like the most relative feature in this model.

**Conclusion**

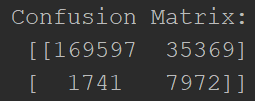
We can check the performance of the models. From the confusion matrix we get each model’s accuracy, precision, recall and F1 Score. However, this performance is based on the testing set. In order to get all the factors, we need to apply our models on the entire data set.

* CEO Names: The precision coefficient is relatively low compared to other coefficient, which means the large number of origin negative samples (the extracted name that is not CEO name) is recognized by model as positive samples. However, the positive-positive samples are large compared to the positive-negative samples (Accuracy Coefficient).



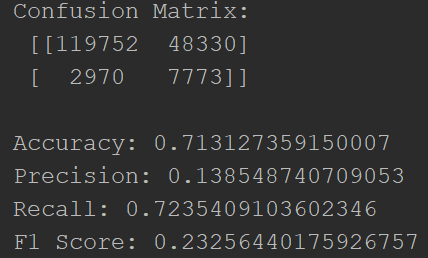
**Results of the testing set**

Then applying our models on the entire dataset, we get 7972 correct CEO names and 35369 false names (total 43341 names) where the whole CEO names are in this corpus.



**Results of the entire dataset**

* Company names: The performance of this model is similar to the CEO names. Precision is low which means there is large number of negative-positive samples in our prediction compared to the positive-positive samples. However, the positive-positive samples are large compared to the positive-negative samples (Accuracy Coefficient).



Then applying our models on the entire dataset, we get 27158 correct company names and 160274 false names (total 187432 names) where the whole company names are in this corpus.

399492 160274

9156 27158

* Percentage: Extract the 58044 aim percentage in the corpus.