**Text Analytics of Business Insider Articles**

Charles Marshall

Prof. Klabjan

IEMS 308

8 March 2020

**Introduction:**

In this project, I performed Natural Language Processing (NLP) on Business Insider articles from the years 2013 and 2014 to extract three different lists:

1. All names of CEOs from the articles
2. All Company names from the articles
3. All percentages in the articles

Business Insider is a financial and business news website, so these categories are common throughout the articles. This project has a focus on regex and creating good features for classification algorithms to extract quality lists of the aforementioned categories.

**Overall Preprocessing:**

The first steps I took in this assignment were to perform some overall preprocessing on the Business Insider corpus. The goal of this preprocessing was to remove fragments of the corpus which would not be useful as features in an NER classifier and not be candidates for extraction for any of the entities. With these fragments removed, it would allow further processing to be quicker and more efficient because there would be less to be considered. With these goals in mind, I decided it was best to remove all Unicode characters and the special character \*. I decided to keep other special characters such as periods, commas, dashes, colons, and percent signs in the corpus because I saw them as crucial for parts of some of our examples candidates. For instance, a company names such as Wall-Mart might have a dash in it and a percentage such as “0.5%” would have both a period and percent sign in it. Next, the corpus was tokenized by sentences. This preprocessing does not pose an inherent performance advantage but is advantageous because individual sentences are easier to examine than individual articles. Lastly, I decided to remove all stop words from the corpus because these are words which are very common throughout the Business Insider corpus yet add very little value in terms of quality features or candidates for extraction. With this in mind, it is best to remove them to most efficiently process the data.

**CEO Classifier:**

**Regex:**

The following regex expression was used to extract CEO candidates from the data set:

**r'[A-Z]\w+ [A-Z]\w+'**

The idea behind this expression is that the vast majority of CEOs in the corpus, as shown by the ceo.csv file, should follow the Firstname Lastname standard for names. In terms of regex, this means that both words should start with capital letters, shown by [A-Z] at the start of each word, and should be followed by letters, shown by \w, the metacharacter for a word character. The plus (+) after the metacharacter means that the word characters repeat. There is a space between the two expressions because all first and last names have a space between them. This regex resulted in 450,405 candidate CEOs.

**Additional Preprocessing:**

No additional preprocessing was performed for the CEO classifier with the thought in mind that the regex expression and chosen features would mainly contribute to extracting CEO names from this point forward.

**Models Considered:**

Three different logistic regression classification models were considered to classify CEO names. The first two models utilized oversampling approaches whereas the third model did not use oversampling techniques. Oversampling techniques were considered because of the vast imbalance in the labels of 1 and 0 when labeling the CEO candidates. Only about 3.4% of the over 450,000 candidates selected from regex were labeled as CEOs (1s). A Synthetic Minority Over-sampling Technique (SMOTE) was used to create synthetic instances of the 1s class in order to increase the sensitivity of the logistic regression model to the minority class, 1s. The difference between the two models utilizing oversampling techniques was that Model 1a utilized the full set of features recommended by RFE whereas Model 1b utilized a subset of the RFE features chosen by trying out different features in order to increase the precision of 1 values for the test data. Model 2 was included in order to compare the performance of a model not utilizing oversampling techniques vs models using oversampling techniques.

**Model Performance:**

Performance of ML algorithms is usually evaluated by precision and recall values for all predicted values. However, in this project, our objective was to extract a list of CEOs, so the most important performance measure was the model’s ability to correctly extract CEOs (True Positives) while not “watering down” the list with false positive predictions. This is conveyed as the precision value of the minority 1 class. The precision values for the minority class of three models over the entire dataset of potential CEOs are as follows:

Model 1a: 0.04

Model 1b: 0.24

Model 2: 0.53

Therefore, Model 2 was selected because of its better relative performance in this measure. Inspection of the extracted list of CEOs for Model 2 versus other models backs up this selection because it generally extracts better quality candidates more frequently than the other models.

**Features Used:**

The model features were selected using Recursive Feature Elimination (RFE) over the entire set of considered features (Appendix 1), then eliminating features which had a p-value greater than 0.05. The features used in this model are as follows:

*caps* – the number of capital letters in the candidate

*ceos\_two* – the word “ceo” in some form is within two words of candidate

*pres\_two* – the word “president” in some form is within two words of candidate

*inv\_two* – the word “investor” in some form is within two words of candidate

*sec\_two* – the word “secretary” in some form is within two words of candidate

*found\_two* – the word “founder” in some form is within two words of candidate

*ceo\_in\_sent* – the word “ceo” in some form is in the same sentence as the candidate

*represent* – the word “representative” in some form is in the same sentence as the candidate

*spok* – the word “spokesman(woman)” in some form is in the same sentence as the candidate

*found* - the word “founder” in some form is in the same sentence as the candidate

All of these features have a p-value less than 0.05, meaning they are all significant features to the model. Additionally*, ceos\_two, pres\_two, inv\_two, found\_two, ceo\_in\_sent*, and *found* all have positive coefficient values, whereas *caps, sec\_two, represent*, and *spok* all have negative coefficient values.

There are three distinct levels of features included in this classifier: word-level, window-level, and sentence-level. It was important to include features from multiple different “levels” in order to try to capture all CEOs. The word level feature, *caps*, was created to eliminate words which have more capital letters than a common name, which usually has two. The window-level features such as *ceos\_two,*are meant to capture when a person’s title is put directly in front or behind their name. In an example sentence this might look like, “First Last is CEO of …” or “Company Name CEO First Last …”. This can be extracted to other titles besides CEO such as Secretary, Investor, etc. Lastly, the sentence-level features such as *ceo\_in\_sent* are meant to capture other instances of titles which are not directly around the name. For instance, in a sentence this might look like, “Tim Cook, who was recently names CEO of Apple.” In this case, Tim Cook is labeled as the CEO in the sentence, but not directly by his name, so the sentence-level feature captures it. Like the window-level feature, this logic can also be used for other titles besides CEO such as Representative and Founder. These titles were picked strategically as either positive or negative indicators of a CEO. For instance, often a CEO of a company also holds the title of President or Founder, but not often will they be labeled with the title of Secretary or Spokesman. These features are included in order to extract cases where CEO is not used to label the CEO of a company in a sentence or a window.

**Model Drawbacks:**

The major drawback with recall of the minority class of predicting 1, which is only 6%. This means that only 6% of the ground truth positives are identified by the model. Comparing the recall of the minority class over the entire candidate dataset:

Model 1a: 0.98

Model 1b: 0.11

Model 2: 0.06

Model 2 is significantly behind Model 1a and only slightly behind Model 1b. However, Model 1a achieves this performance by having a tendency to predict mainly 1s and not as many 0s, leading to major precision issues expressed in the Model Performance section, and a general poor-quality extracted list of CEOs. According to the confusion matrix in Figure 1, there are 14,496 candidates which are false negatives. My contention is that the model is not able to correctly predict these candidates because they are not identified by the features, so creation and use of better features in the future would help to extract more of these values and increase recall.

A screenshot of a cell phone

Description automatically generated

Figure 1: Confusion Matrix for Model 2 for CEO extraction

**Company Classifier:**

**Regex:**

The following regex expression was used to extract company candidates from the data set:

**r'(([A-Z][A-Za-z0-9]+[ -]?)+)'**

The idea behind this expression is that the vast majority of companies in the corpus, as shown by the company.csv file, should follow the form where they are every word should be capitalized. The company names can be a minimum of one word, but there is no maximum number of words in a company name. In terms of regex, this means that all words should start with capital letters, shown by [A-Z] at the start of the expression. Additionally, a company name is generally filled with either letters of numbers repeating until the end of the word, shown by [A-Za-z0-9]+ The plus (+) after this part means that the word characters repeat until a space or dash, [ -]. The terminating condition is that the next word does not start with a capital letter. This regex resulted in 1,148,996 candidate companies.

**Additional Preprocessing:**

The regular expression tended to output candidates in which the last character in the string was a space. This effected the ability to match candidates with the company.csv list to create labels. In order to deal with this, a function was written to get rid of the last character for a candidate if that last character is a space. This vastly improved the number of candidates which were labeled.

**Models Considered:**

Three different logistic regression classification models were considered to classify CEO names. The first two models did not use oversampling techniques whereas the third model utilized an oversampling approach. The reasoning behind utilizing SMOTE was explained in the CEO section and the can be applied here as well because imbalance in the labeled values seen by the proportion of labeled 1s, the minority class, to 0s, the majority class being around 8.9%. The difference between the two models not utilizing oversampling techniques was that Model 1 selected features based on p-values from the full set of features whereas Model 2 utilized features suggested by RFE. Model 3 was included in order to compare the performance of the two models not utilizing oversampling techniques vs a model using an oversampling technique. Model 3 utilized a subset of the RFE features chosen by trying out different features in order to increase the precision of predicted company values for the test data.

**Model Performance:**

As explained in the CEO section, in this project, the model’s performance was best conveyed by its precision in predicting companies (1s) over the full dataset of candidates. The precision values for the minority class of three models over the entire dataset of potential companies are as follows:

Model 1: 0.48

Model 2: 0.54

Model 3: 0.47

Therefore, all models performed similarly in this measure, with Model 2 being slightly the best. However, Model 3 was selected because it achieved this performance while correctly predicting about 3 times as true positive values as the other two models. Inspection of the extracted list of companies for Model 3 versus other models backs up this selection there are companies included in Model 3’s list which were not picked up by the other two models.

**Features Used:**

The model features were first selected using Recursive Feature Elimination (RFE) over the entire set of considered features for companies (Appendix 2). Then, a subset of these features was chosen by manually trying different features to maximize precision of the predicted companies. The features used in this model are as follows:

*corp* – a form of the word “Corporation” is included in the candidate name (including Corp)

*group* –the word “Group” is included in the candidate name

*inc* – the word “Inc” is included in the candidate name

*company* – the word “Company” is included in the candidate name

*association* – the word “Association” is included in the candidate name

*foundation* – the word “Foundation” is included in the candidate name

All of these features have a p-value less than 0.05, meaning they are all significant features to the model. Additionally*, corp, group,* and *inc* all have positive coefficient values, whereas *company, association,* and *foundation* all have negative coefficient values.

Similarly, to the features for the CEO classifier, features of multiple “levels” were considered for the algorithm which can be seen in Appendix 2. However, feature selection through model performance led to only the selection of candidate-level features in the company classifier. All of the features included check to see if a particular word is included within the candidate companies’ name. These feature words were chosen strategically to either indicate a company or not a company. For instance, the words Association and Foundation are usually not used when naming a company, and intuitively have negative coefficient values in the model. On the other hand, words such as Corp, Inc and Group are often used in business names, and intuitively have positive coefficients in the model.

**Model Drawbacks:**

The major drawback with recall of the minority class of predicting companies (1s) on the entire dataset, which is only 3%. This means that only 3% of the ground truth positives are identified by the model. Comparing the recall of the minority class over the entire candidate dataset:

Model 1: 0.01

Model 2: 0.01

Model 3: 0.03

Model 3 performs better than both Models 1 and 2 in this metric. However, according to the confusion matrix for Model 3 in Figure 2, there are 100,005 false negative values, companies which were not identified by the model. Similarly, to the CEOs model, my contention is that the model is not able to correctly predict these candidates because they are not identified by the features, so creation and use of better features in the future would help to extract more of these values and increase recall.

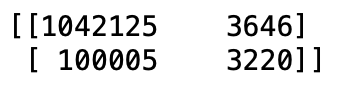


Figure 2: Confusion Matrix for Model 3 for company extraction

**Percentages Classifier:**

**Regex:**

Three different regex expression were used to extract percentage candidates from the data set because of the variety of different ways percentages can appear. The first expression is as follows:

**r'\d\*\.?\d+'**

The idea behind this regular expression is that it would extract all numerical and decimal numerical candidates from the corpus. For instance, the numbers 10, 5 and 0.5 would both be extracted using this rule. It mainly does this using \d, which is the metacharacter for a digit. The decimal in the expression is optional as shown by the question mark next to it.

The second expression is:

**r'one[\s|-]?hundred|fourteen|fifteen|sixteen|seventeen|eighteen|nineteen|zero|one|two|three|four|five|six|seven|eight|nine|ten|eleven|twelve|thirteen'**

The second expression extracts the set of written out numbers one through nineteen and one hundred. It utilizes a brute force method by comparing to a “list” of these written out numbers by using the regex “|” which is an or operation.

The third expression is:

**r'((twenty|thirty|fourty|fifty|sixty|seventy|eighty|ninety)(\s|-)?(one|two|three|four|five|six|seven|eight|nine)?)'**

The third expression utilizes the pattern displayed by numbers greater than or equal to 20 when written out of stating the tens digit first then the ones digit. When written out these digits are commonly connected by a dash, space, or linked together, expressed in regex as (\s|-)?. This expression extracts all written out numbers from twenty up to one hundred. Since the vast majority of percentages are values between zero and one hundred, this should be sufficient.

The idea behind the combination of these regex expressions is to extract all numbers, either in digit form or written out as shown by the percentage.csv file, which are included in the corpus. After extracting all numbers, features in the classification algorithm should differentiate between percentages and regular numbers. This regex resulted in 591,173 candidate percentages.

**Additional Preprocessing:**

The output of the third regular expression is a list, so an additional line of preprocessing was used to extract only the first value of this list, which is the full number included in the corpus.

**Models Considered:**

Only one model was considered for the extraction of percentages. Other models were not considered because of the general good performance of this model. The utilization of oversampling techniques was not appropriate because there was not an imbalance between the two labeled classes. Overall, about 42.5% of candidate percentages were labeled as percentages in preprocessing.

**Model Performance:**

As explained in the CEO section, in this project, the model’s performance was best conveyed by its precision in predicting percentages (1s) over the full dataset of candidates. The precision value for predicting percentages of the model over the entire dataset of potential companies is 0.69. My contention is that this value is artificially deflated because there are some candidates which are labeled as non-percentages because they do not show up in the percentage.csv dataset but are correctly predicted by the model as percentages. There could be up to 23,449 of these cases according to the confusion matrix in Figure 3. The recall of the model when predicting percentages is 0.21. This is a relatively good value compared to other models in this project, but not an overall great value. However, I believe this value is also artificially deflated by the inclusion of a large set of numbers which are not percentages in the percentage.csv dataset. This inclusion is therefore labeling some non-percentage numbers in the corpus as percentages (1s). The model, via the features, correctly classifies these numbers as non-percentages, but because they were labeled as percentages, they are considered False Negatives. Therefore, the recall value should be higher.

A close up of a logo

Description automatically generated

Figure 3: Confusion Matrix for Percentage extraction model

**Features Used:**

The model features were created using functions which consider both the number in the sentence and the number on its own. It was confirmed that both features have p-values less than 0.05, and therefore are significant. The features used in this model are as follows:

*year* – whether the number is potentially a year

*perc* –whether there is a percent sign (%) or some form of the word “percent” directly following the number

*year* has a negative coefficient value whereas *perc* has a positive coefficient value. The year feature was used to eliminate a common number in the corpus which was obviously not a percentage, a year. Intuitively it is a negative coefficient within the model. The most powerful coefficient in the model is *perc,* which is designed to capture all forms of a percent indicator next to the candidate such as %, “percent”, and “percentage”. It is important in eliminating numbers which are not years, but not percentages from consideration.

**Model Drawbacks:**

The model has no major drawbacks and generally does a good job classifying percentages in the corpus. Precision and recall values from this model should be taken with a grain of salt according to the reasoning explained in the Model Performance section.

**Appendix 1 – All Potential Features in CEO Classifier**

*Length –* length of the candidate in terms of the number of characters

*sent\_len –* length of the sentence in terms of number of characters

*caps –* number of capital letters in the CEO candidate

*sent\_caps –* number of capital letters in the sentence

*who -* the word “who” in some form is within two words of candidate

*ceos\_two* – the word “ceo” in some form is within two words of candidate

*sen\_two -* the word “senator” in some form is within two words of candidate

*pres\_two* – the word “president” in some form is within two words of candidate

*inv\_two* – the word “investor” in some form is within two words of candidate

*aut\_two -* the word “author” in some form is within two words of candidate

*rep\_two -* the word “representative” in some form is within two words of candidate

*amb\_two -* the word “ambassador” in some form is within two words of candidate

*sec\_two* – the word “secretary” in some form is within two words of candidate

*exp\_two -* the word “expert” in some form is within two words of candidate

*spoke\_two -* the word “spokesman(woman)” in some form is within two words of candidate

*gov\_two -* the word “governor” in some form is within two words of candidate

*part\_two -* the word “partner” in some form is within two words of candidate

*found\_two* – the word “founder” in some form is within two words of candidate

*ceo\_in\_sent* – the word “ceo” in some form is in the same sentence as the candidate

*sens -* the word “senator” in some form is in the same sentence as the candidate

*pres -* the word “president” in some form is in the same sentence as the candidate

*inv -* the word “investor” in some form is in the same sentence as the candidate

*aut -* the word “author” in some form is in the same sentence as the candidate

*represent* – the word “representative” in some form is in the same sentence as the candidate

*ambass -* the word “ambassador” in some form is in the same sentence as the candidate

*secr -* the word “secretary” in some form is in the same sentence as the candidate

*exp -* the word “expert” in some form is in the same sentence as the candidate

*spok* – the word “spokesman(woman)” in some form is in the same sentence as the candidate

*gov -* the word “governor” in some form is in the same sentence as the candidate

*part -* the word “partner” in some form is in the same sentence as the candidate

*found* - the word “founder” in some form is in the same sentence as the candidate

*who\_in\_sent -* the word “who” in some form is in the same sentence as the candidate

**Appendix 2 – All Potential Features in Company Classifier**

*comp\_in\_sent –* the word “Company” is included in the sentence in which the candidate appears

*stock -* the word “stock” is included in the sentence in which the candidate appears

*shares -* the word “shares” is included in the sentence in which the candidate appears

*trade -* the word “trade” is included in the sentence in which the candidate appears

*length –* the length in number of characters of the candidate

*plural –* whether the candidate word is plural or not

*number\_words –* the number of words in the candidate

*location –* whether the candidate is located at the start of its sentence or not

*corp -* a form of the word “Corporation” is included in the candidate name (including Corp)

*group -* the word “Group” is included in the candidate name

*holding -* the word “Holding” is included in the candidate name

*inc –* a form of the word “Inc.” is included in the candidate name

*company -* the word “Company” is included in the candidate name

*association -* the word “Association” is included in the candidate name

*foundation -* the word “Foundation” is included in the candidate name