IEMS 308
Assignment 3
Northwestern University

Word Extraction

Bruno Zecchi

For this assignment we were tasked with extracting a list of CEO names, company names, and percentages from a record of news articles published each day of 2013 and 2014. We took given training data, built negative samples, and developed a logistic regression classification model that assigned a token to one of four categories: CEO name, company name, percentage, or other. Seven features were used that analyzed the characteristics of the token's characters, as well as checking whether a token was inside one of the prebuilt lists of the three entities. We checked the model on a testing dataset and determined it to be highly accurate (90% +). Lastly, we applied the model to each of the individual articles and extracted their relevant entities. The model appears to be accurate with percentages and company names, but further work is required in extracting CEO names. Further features need to be included that help the model better distinguish CEO names from regular names. A more detailed explanation of the analysis and model evaluation can be seen below.

Analysis Process

Data Cleaning: The given training labeled data is imported and cleaned. All duplicates and excess space characters are removed.

Building Integrated Dataframe: We combined the 3 different given labels into a single dataframe, and provided the label type (CEO-1, company-2, or percentage-3) into a new column called "Type".

Index	Name	Type
1758	99 percent	3
2403	15900%	3
272	Jacobs	1
5946	Service Co Ltd	2
2191	1.75 percent	3

Form Negative Samples: Data points are taken from the internet and will act as negative samples when building the classification model. Negative samples include **names of famous people, fictional company names, locations, and random English words**.

Name	Type
Martin Lawrence	4
Texas	4
24.041	4
57.6	4

Negative Samples Dataframe

Building Final Dataframe: The real labels and negative samples are mixed up and combined into a final dataframe. The "Type" column is added so that testing of a model can be done later.

Develop Features: Each token of the final dataframe is given a 1/0 value for the seven features developed. The features are described below:

- Feature 1: Is token in prebuilt lists of CEO names?
  - Is it a last name?
  - Is a mix of this token and the next three tokens a full name in list?
- Feature 2: Is token in prebuilt list of percentages?
- Feature 3: Is token in prebuilt list of companies?
  - Is a mix of this token and the next three tokens a full name in list?
- Feature 4: Is the next token "%" or "percent"?
- Feature 5: Is token's first letter capitalized?
- Feature 6: Is current token and next token capitalized?
- Feature 7: Do next three tokens contain "Co", "Co.", "Inc", etc ?

Index	Name	Type	var1	var2	var3	var4	var5	var6	var7
0	99	3	0	0	0	1	0	0	0
1	percent	3	0	0	0	1	0	0	0
2	clearly	4	0	0	0	0	0	0	0
3	leon	4	0	0	0	0	0	0	0
4	unwrap	4	0	0	0	0	0	0	0

Build Classification Model: We use a logit model that takes the seven features and the "Type" column of a training subset to build a classifier. We then test the model on a testing subset and evaluate its accuracy. The results of the evaluation are shown below:

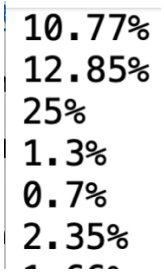
	Percentage Predicted
Total	97%
CEO Names	97%
Company Names	90%
Other	97%
Incorrect Predictions	3%

It is worth noting that the performance of the model is hard to measure at this point. Testing dataset comes from the same source of the training dataset, so the first three features will be very accurate. They might not be as accurate with outside data.

Apply Model to Text Files: We build a for-loop that opens each of the articles and identifies all CEO names, company names, and percentages in it. Those values are appended to a final list for each corresponding type. Those final lists now contain those three types of labels in all articles. The three entities extracted are displayed in the 3 given txt files: finalCEO.txt, finalCompany.txt, and finalPercentage.txt.

Model Performance & Future Development

By analyzing the text files, we see that the model does a good job of extracting percentages from the articles.



Percentages Extract

Company Name Extracts

CEO Names Extracts

We also see that the model also did a good job of extracting company names from the articles. However, it seemed less accurate in extracting CEO names. Several non-CEO names appear throughout the file, as well as location and company names. This fall in performance could be linked to the fact that it is hard to distinguish a CEO name form a regular name. Many of the features will be the exact same for CEO and non-CEO name. The difference can only be seen by researching the name (something our model cannot do) or comparing a name to a prebuilt list. If a given CEO name is not in the prebuilt list or is written in a different format, the model will have a hard time understanding what it is. Further developments of the model should involve adding features that are able to distinguish CEO and non-CEO names better.