Project 3 Methodology

IEMS 308 – Northwestern University

Jake Atlas

Data Preprocessing

The corpus of text files was first read in and combined. The NLTK package in Python was used for sentence segmentation and word tokenization. The text was then normalized through the removal of unnecessary punctuation. In order to eliminate the unnecessary words, the text was compared to a list of stop words, and all instances of these words were removed from the text. Next, part of speech tagging was used, which enabled lemmatization. Lemmatization was selected in place of stemming because it was found that stemming the text turned many proper names lowercase, which would have inhibited later analysis.

The supplied training data was preprocessed analogously. In order to find instances of the training data in the corpus, it was important to apply the same preprocessing methods to the training data that were applied to the corpus.

Regular Expressions for Training Data

Each record in the preprocessed training data was treated as a regular expression. Those training set elements for which matches were found in the corpus were added to one of three lists for CEOs, companies, or percentages.

Constructing Feature Matrix for Training Model

To build models that are capable of identifying CEOs, companies, and percents, it was important to add a set of negative samples to the data. To do this, the NER tagger in NLTK was used to identify entities that were people and organizations.

Using manual inspection, the CEOs and companies were removed from these lists, respectively, and the remaining elements were used to balance the set of positives in the training data supplied. The following feature matrices were then created:

CEOs

1 if "CEO" appears nearby in the corpus,; 0 otherwise	Number of characters in the name	Number of capitalized letters in the name	Number of "words" in the name

Companies:

1 if "Company," "Inc," or "Corp" appear nearby in the corpus; 0 otherwise	Number of characters in the name	Number of capitalized letters in the name	Number of "words" in the name	1 if the name contains any of: Group, LTD, Airline, LL, Management, Capital, Advisors, Partner, LP, or Associate; 0 otherwise

Percents:

1 if "percent" or "%" appears nearby; 0 otherwise

The negative samples were replicated until the ratio of positive samples to negative samples was slightly better than 70:30, and then a logistic regression model was trained on this data.

Regular Expressions for Test Data

In order to prevent from having to loop through each unigram, bigram, trigram, and 4-gram, it was necessary to subset the text using regular expressions. The following regular expression was used to identify possible percents:

'\s([0-9]+|[a-zA-Z]+-?[a-zA-Z]*|[0-9]+\.[0-9]+)\s?(%|percent)(age point)?'

By locating all text corresponding to this regular expression, it was no longer necessary to create a new row in the feature matrix for each m-gram. Rather, the features only had to be calculated for expressions matching the regular expression.

This same logic was applied to CEOs and companies. The same regular expression was used for both CEO and company subsetting:

(?<!\.\s)[A-Z][a-z]+(?:\s?[A-Z]?\s?[A-Z][a-z]+)*

Once a set of potential CEOs and companies was identified using this regular expression, a feature table could be constructed for only the set of phrases identified instead of for all m-grams in the corpus.

Applying the Classification Model to the Test Set

After the feature matrices were created, the models trained on the training set were applied to the test sets for CEOs, companies and percentages. The entities with probabilities meeting a threshold were selected to enter the final list.

Overall Performance

The percentages model appears to have worked quite well. The p-value associated with the primary predictor in the model is essentially 0. The model does capture any percents that are phrased using some variety of the word "percent" or the percentage symbol, %. Numbers that are representative of percents but do not have a direct reference to percents nearby are likely to not be noticed by the model and therefore not included in the list.

The CEO and company models appear to have worked to some degree, but lack the ability to distinguish more ambiguous cases. The z-scores associated with the predictors in the models range from quite good (nearly 0) to somewhat high, at 0.3. However, since the goal is to identify CEO and company names and not to create models with all statistically significant coefficients, I have allowed all the predictors to remain in the models. To improve these models, a weighting scheme could be introduced that more heavily favors certain predictors. It is clear from the lists of CEOs and companies found that the model has room for improvement, but it is certainly not ineffective.

In all cases, adjusted R^2 values corroborate these conclusions. The classifiers built manage to account for approximately one-third of the variation in classification as a percent, company or CEO. The model as a whole, however, involves subsetting based on regular expressions prior to the application of logistic regression models. Consequently, the process takes more details into account and should therefore classify correctly for well over the one-third benchmark.