Text Inference

Enzo Profli

1 CEOs

The first Name Entity Recognition (NER) model was run to assess CEO names in the Business Insider data. The logistic regression model used 2013 articles as a training dataset, and then subsequently tested on 2014 data - the data was segmented into sentences to allow for easier matching. In business journalism (and by inspection), CEOs are usually denominated using a *FirstName LastName* approach. Thus, the CEO labels were altered to match this pattern (without commas, or *LastName FirstName*, etc.). Moreover, the datasets were generated by filtering for consecutive capitalized words, and they have two columns: a Sentence column containing the match plus 3 preceeding words and 3 word succeeding words, and a Person column containing the match. If the Person column matches any of the CEO labels, they are assigned as a ceo = 1 variable, and 0 otherwise. Finally, the re.findall function in Python was used to match the following regex in the data: '((?:\S+\s+){0,3}\b([A-Z]+[a-z]*(?=\s[A-Z])(?:\s[A-Z]+[a-z]*){1})\b\s*(?:\S+\b\s*){0,3})'.

Given this structure, features were generated to train the model, based on Sentence column string patterns:

- name: is the first capitalized word a common name? To produce this feature, the babynames dataset in R was utilized to generate a complete list of babynames, using Census data.
- location: does the column match a location? The world.cities dataset in R was utilized to produce matches.
- location2: do any of the words in Person contain a location identifier? Examples: Street, Place, Square, River, etc.
- company: any company identifier? Examples: Corp, Ltd, etc.
- company2: is there a company label in the sentence? This feature was generated by matching with provided company labels.
- dow: any day of the week identifier in Person?
- n_capitalized: number of capitalized words in sentence.
- denomination: CEO name usually followed by position description (CEO, chief, leader, president, billionaire, etc.). Are there any of those identifiers in the sentence?

Given these features, the model was trained, with good results. The model's area under the curve (AUC) was 0.8, and the ROC curve can be seen below in Figure 1. The highest probability matches (over a .2 probability match in the 2014 data) can be seen in ceo_matches.csv.

```
## Area under the curve: 0.7843
```

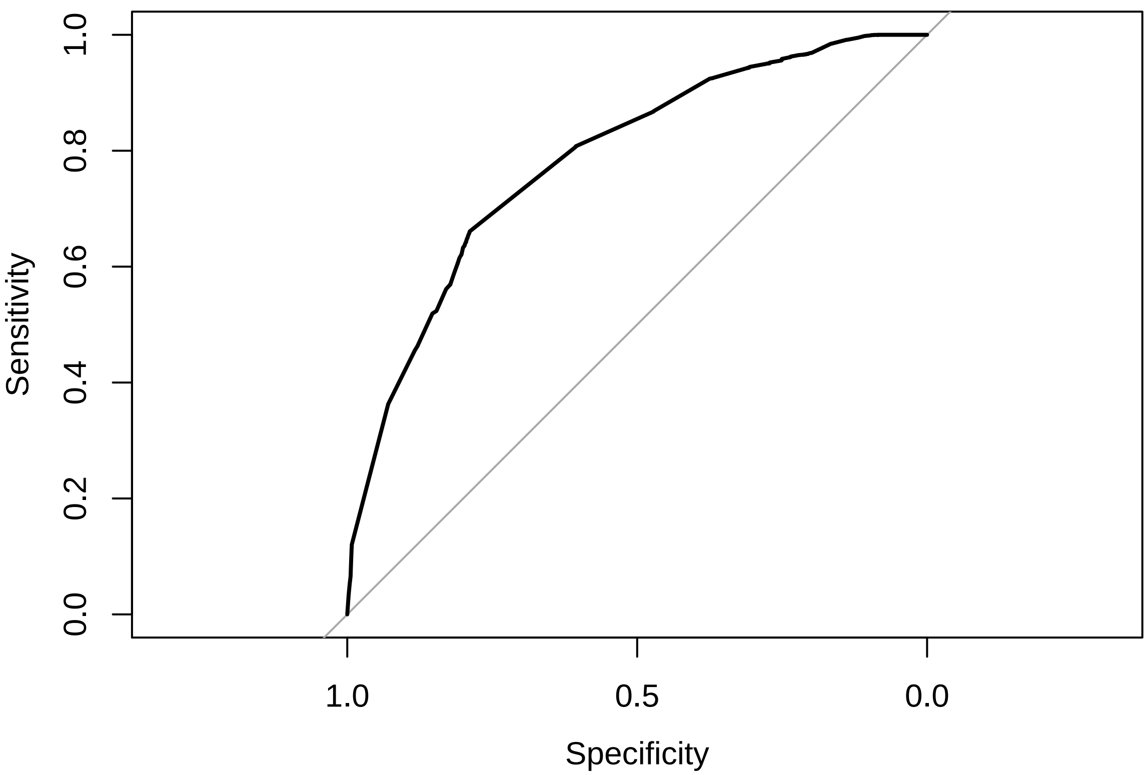

Figure 1: ROC Curve for CEO NER model

2 Companies

A similar approach was conducted to find companies, with slightly lower performance. Only the first two names of company labels were kept, and words such as Co, Corp, Group, etc. were removed. Companies may have one or multiple names, and so the regex was altered to match this situation: ((?:\S+\s+){0,3}[ˆ\\.]\b(?:[A-Z][a-z]*\s+)+\b\s*(?:\S+\b\s*){0,3}) - note that we still keep words around the match, as with CEOs. We trained the model with similar variables as above, but with some additions:

- stock: in these articles, companies are usually mentioned in the context of the stock market. Does the sentence allude to stocks?
- verbs: similar to stock, but with verb stems (earn, jump, stumble, etc.).
- political: a lot of articles are related to politics, and to filter these out, we ask: are there politically-related words in the sentence?
- preposition: by inspection, many company names are preceded by prepositions (ex: "at Tesla, engineers. . . ", or "talks with Tesla have broken down"). Do any prepositions precede the potential company name?
- possession/comma: similar to preposition, but with possession identifiers (ex: "Tesla's IPO") and commas (ex: "Tesla, whose shares jumped,. . . ")

With these variables, the model achieved an AUC of .69, which means the NER model is somewhat predictive, but prone to errors. By inspection, companies are mistaken for countries, which possibly relates to the fact that the labels themselves are not perfect (for example, “United States” is labeled as a company, and thus countries/nationalities are probably mistaken for companies). Figure 2 displays the ROC curve for this model, and company_matches.csv shows matches that were considered very likely to be companies (probability > 0.35).

```
## Area under the curve: 0.691
```

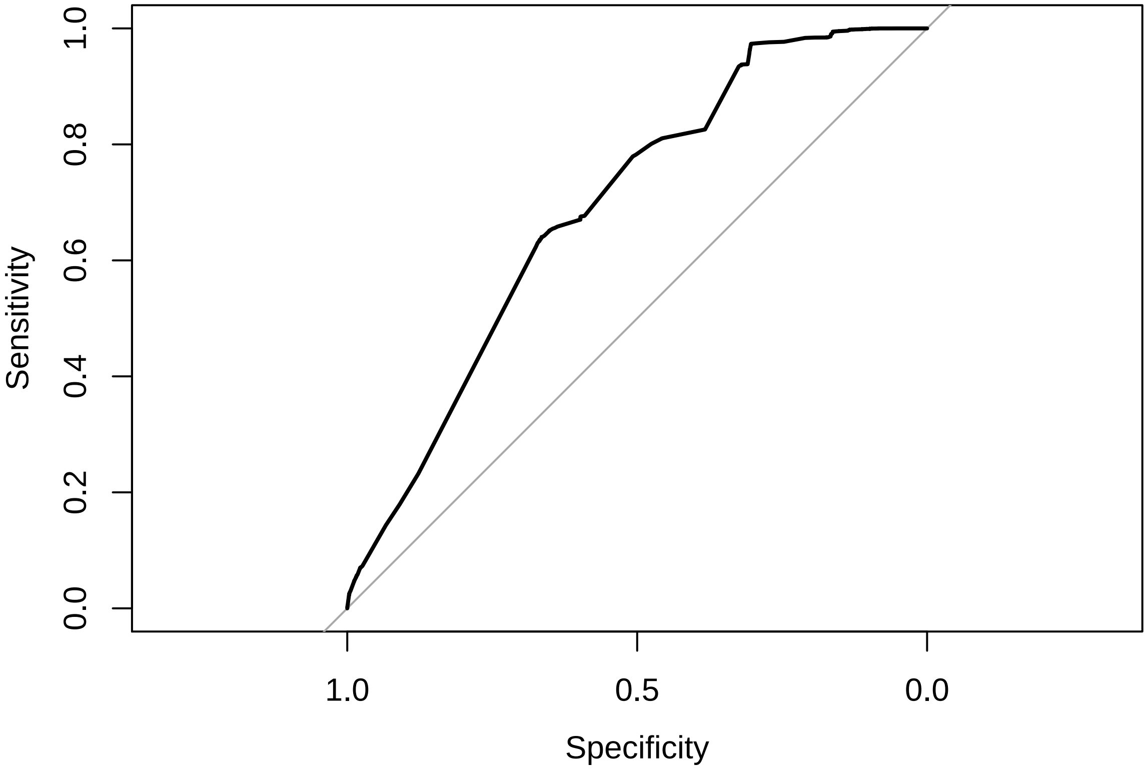

Figure 2: ROC Curve for company NER model

3 Percentages

A different approach was made to subset percentages. As numbers can vary widely, I thought it better not to use labels, and use broad regex patterns instead, to catch a wide array of percentage expressions. So, in this case, no NER model was used. Instead, the following regex was used: ((?:\S+\s+){0,3}\b(\S{1,5}%|(?:[0-9]{1,3}.*[0-9]{0,2}|half a*|quarter a*|\S{1,10}) (?:percent|percentage points*))\b\s*(?:\S+\b\s*){0,3})|. It matches sentences around the expression, as previously, but returns the match in a separate column. This returns x% as well as written declarations (five percent), or more peculiar expressions (a quarter percentage points). The list of distinct percent expressions matched can be seen in percentage_matches.csv.