Brent Claypool
IEMS 308
March 3 2021

HW 3

Methodology:

For this assignment, I wanted to establish features for a classification model. I created the following classification features in Excel: ‘is_alpha’, ‘is_numeric’, ‘capital_initial’, ‘unigram’,’bigram’, ‘trigram’, ‘text_length’, ‘contain_period’, ‘contain_percent’, ‘contain_hyphen’. Description of features are as follows:

Feature	Excel Formula	Description
‘is_alpha’	=IF(ISTEXT(A2),1,0)	This formula returns 1 if the text contains alpha characters, 0 otherwise
‘is_numeric’	=IF(ISNUMBER(A2),1,0)	This formula returns 1 if the text is completely numeric, 0 otherwise
‘capital_initial’	=IF(ISTEXT(A2),IF(EXACT(LEFT(A2,1),PROPER(LEFT(A2,1))),1,0),0)	This formula returns 1 if the text is alpha and the first letter is capitalized, 0 otherwise
‘unigram’	=IF(LEN(A2) - LEN(SUBSTITUTE(A2," ",""))= 0,1,0)	This formula returns 1 if the text is a unigram, 0 otherwise
‘bigram’	=IF(LEN(A2) - LEN(SUBSTITUTE(A2," ",""))= 1,1,0)	This formula returns 1 if the text is a bigram, 0 otherwise
‘trigram’	=IF(LEN(A2) - LEN(SUBSTITUTE(A2," ",""))> 1,1,0)	This formula returns 1 if the text is a trigram or contains more than 3 words, 0 otherwise
‘text_length’	=LEN(A2)	This formula returns the number of characters in the text
‘contain_period’	=IF(LEN(A2) - LEN(SUBSTITUTE(A2,".",""))> 0,1,0)	This formula returns 1 if text contains “.”, 0 otherwise
‘contain_hyphen’	=IF(LEN(A2) - LEN(SUBSTITUTE(A2,"-",""))> 0,1,0)	This formula returns 1 if text contains “-”, 0 otherwise
‘contain_percent’	=IF(LEN(A2) - LEN(SUBSTITUTE(A2,"%",""))> 0,1,0)	This formula returns 1 if text contains “%”, 0 otherwise

‘contain_word_percent’	=IF(LEN(A2) - LEN(SUBSTITUTE(A2,"percent",""")) > 0,1,0)	This formula returns 1 if text contains “percent”, 0 otherwise

In the lecture, the professor mentioned it was good practice to include a feature which described the tag of the previous or next token, but for the given CEO, Company, and Percent data, we do not have information about the surrounding tokens, so we cannot use this feature.

To incorporate the article data, I used SpaCy’s NER model. The function nlp() combines the preprocessing steps. It covers, sentence segmentation, tokenization, remove stop words, and normalization, then it assigns part-of-speech tags and text categories. I ran nlp() on all text articles to find their text labels. I then added a small sample of these tokens to my running Excel file that contains the CEO, Company, and Percent categories to serve as ‘negative’ samples. I included this result dataset in the Github submission. I trained my classification model on a split of 75% training and 25% test.

Results:

For the results of the random 5 sampled articles,

```
print(y_pred)
```

```
0       Company
1       Percent
2       Company
3       Company
4           CEO
        ...
4395        CEO
4396        CEO
4397    Company
4398    Percent
4399    Percent
Length: 4400, dtype: object
```

The full results can be found by running my code at the end, my laptop could not output all of the responses for the 720 articles.