\_README\_

Technical Report for Final Model

Authors: Connor Allison, Nicholas Mata, Joel Fry, Royalpreisthood Olola

IS 6713

For this project we were tasked with creating two models: one to predict if a tweet is related to Technology and one to predict if the tweet is related to Accessibility. For both the Technology and Accessibility data, we started by reading in the data we had annotated and randomly splitting it into training, validation and testing sets. Of the 1000 datapoints, 70% went to training, 15% to validation and 15% to testing.

Next, we developed a lexicon classifier class that reads in a list of accessibility words and a list of technology related words. It can make a prediction about the tweet based on whether the tweet contains any of the related words. It can also return a count of technology related and accessibility related words.

Then we used this lexicon classifier to append a list of lexicon features to the tweets. We were able to run a model for each category using just these lists however this alone did not make the best predictions. I will get to the exact numbers shortly.

Using a Vectorizer, we next fit and transformed the X\_data, converted the lexicon feature data to numpy arrays, and used hstack to combine them. Testing our models with different vectorizers eventually led us to conclude that the TfidVectorizer with ngram\_range = (1,1) yields the best results for our technology models. Alternatively, the CountVectorizer with parameters ngram\_range = (1,2), min\_df = 3, max\_df = 0.95, yields the best results for our Accessibility data. At this point we had everything in place to create and tweak predictive models.

We ran several different models on the validation set for both the Accessibility and Technology data. For the accessibility data, we adjusted the ‘C’ parameter, ‘penalty’ parameter, ‘stop\_words’, min\_df, max\_iter, lowercase, ngram\_range, refit, and class\_weight parameters.

Through trial and error we were able to conclude the best model resulted from using max\_iter = 10000, dual = False, tol=1e-4, 'C': [0.01, 0.1, 0.5, 1, 2, 10], 'penalty': ['l1', 'l2'], ngram\_range = (1,2), min\_df = 3, max\_df = 0.95, StratifiedKFold(n\_splits=5), and refit = True. Using different combinations of these parameter we ran models using several different predictors.

The predictors tested were X\_acc\_val2 (The tweet without any lexicon features), X\_acc\_val\_lexicon\_features (just the lexicon features without the tweet itself), X\_acc\_val\_w\_lex(the tweet and lexicon feature hstacked).

To evaluate each of these models we used Micro and Macro Precision, Recall, F1 as well as misclassification rate. Precision measures the accuracy of positive predictions, that means out of all the positive predictions it made, what percentage was correct (macro precision gives equal weight to each class while micro looks at the total true positives across all classes). Recall measures the ability of the model to find all the positives. Finally, F1 provides a single score that balances precision and recall.

For the X\_acc\_val2 model the results were:

Acc Val Macro Precision: 0.8015

Acc Val Macro Recall: 0.7759

Acc Val Macro F1: 0.7855

Acc Val Micro Precision: 0.8133

Acc Val Micro Recall: 0.8133

Acc Val Micro F1: 0.8133

Acc Val Misclassification Rate: 0.1867

For the X\_acc\_val\_lexicon\_features model:

Acc Val Macro Precision: 0.7649

Acc Val Macro Recall: 0.6572

Acc Val Macro F1: 0.6652

Acc Val Micro Precision: 0.7467

Acc Val Micro Recall: 0.7467

Acc Val Micro F1: 0.7467

Acc Val Misclassification Rate: 0.2533

For the X\_acc\_val\_w\_lex model:

Acc Val Macro Precision: 0.8053

Acc Val Macro Recall: 0.7900

Acc Val Macro F1: 0.7965

Acc Val Micro Precision: 0.8200

Acc Val Micro Recall: 0.8200

Acc Val Micro F1: 0.8200

Acc Val Misclassification Rate: 0.1800

Our best predictor of Accessibility as indicated by all out our evaluative measures, was the model that took the tweets themselves and their lexicon features into consideration. Thus, we used this combination of features for our final model and ran our test dataset.

Acc Test Macro Precision: 0.8191

Acc Test Macro Recall: 0.7952

Acc Test Macro F1: 0.8055

Acc Test Micro Precision: 0.8467

Acc Test Micro Recall: 0.8467

Acc Test Micro F1: 0.8467

Acc Test Misclassification Rate: 0.1533

We also used a confusion matrix to explore the misclassification rate further. We wanted to see how each class was being misclassified.

Misclassification Rates per Acc Class:

Acc Class 0: 0.0841

Acc Class 1: 0.3256

Here we see that the model was misclassifying Accessibility related tweets as non-accessibility related at a much higher rate than it misclassified non-accessibility tweets as accessibility related

We used the same process for the technology data. The most of the best parameters remained the same with the notable exception of TfidfVectorizer(ngram\_range = (1,1)) being much better then CountVectorizer for the tech data. The resultd for the different predictors are as follows:

X\_tech\_val\_lexicon\_features:

Tech Val Macro Precision: 0.8013

Tech Val Macro Recall: 0.8030

Tech Val Macro F1: 0.7999

Tech Val Micro Precision: 0.8000

Tech Val Micro Recall: 0.8000

Tech Val Micro F1: 0.8000

Tech Val Misclassification Rate: 0.2000

X\_tech\_val2:

Tech Val Macro Precision: 0.8347

Tech Val Macro Recall: 0.8331

Tech Val Macro F1: 0.8266

Tech Val Micro Precision: 0.8267

Tech Val Micro Recall: 0.8267

Tech Val Micro F1: 0.8267

Tech Val Misclassification Rate: 0.1733

X\_tech\_val\_w\_lex:

Tech Val Macro Precision: 0.8160

Tech Val Macro Recall: 0.8135

Tech Val Macro F1: 0.8066

Tech Val Micro Precision: 0.8067

Tech Val Micro Recall: 0.8067

Tech Val Micro F1: 0.8067

Tech Val Misclassification Rate: 0.1933

Our best predictor of Technology as indicated by all out our evaluative measures, was the model that took the tweets themselves and not their lexicon features into consideration. However, when we ran our test data using the addition of lexicon features, it outperformed the model making predictions with just the tweets. Thus, we ran our final model with the addition of lexicon features:

X\_tech\_test\_w\_lex:

Tech Test Macro Precision: 0.8071

Tech Test Macro Recall: 0.8167

Tech Test Macro F1: 0.8108

Tech Test Micro Precision: 0.8200

Tech Test Micro Recall: 0.8200

Tech Test Micro F1: 0.8200

Tech Test Misclassification Rate: 0.1800

We also used a confusion matrix to explore the misclassification rate further. We wanted to see how each class was being misclassified.

Misclassification Rates per Tech Class:

Tech Class 0: 0.1702

Tech Class 1: 0.1964

For our tech data the two classes had very similar misclassification rates.

Things to improve in the future:

* More comprehensive lists of related words
* Larger dataset so that we can have more training
* Additional lexicon features (Note: I tried adding more weight to certain words, length of tweets and average length of each word in the tweet but these all made the model worse)
* Improve gold standard