

Submission 1

I used average perceptron on the dataset of spacy-embeddings features.

Statistics

Here is the statistics of the labels in the dataset:

```
===== train data statistics=====
{1.0: 3529, -1.0: 3559}
===== test data statistics=====
{1.0: 754, -1.0: 765}
```

I converted every label 0 to -1 for the perceptron. If we always predict the label as the most common label, test accuracy = 49.8% and train accuracy = 50.2%. The learned model should perform better than this baseline.

5-fold cross validation

Hyperparameters' values:

learn_rates = [1,0.1,0.01]

margins = [1,0.1,0.01]

I trained the perceptron with learning rate decay and in 10 epochs for each validation fold.

best margin = 0.1

best learning rate = 0.1

mean accuracy of best param: 0.7444977983639309

I trained the perceptron with learning rate decay and in 20 epochs for each validation fold.

best margin = 1

best learning rate = 0.01

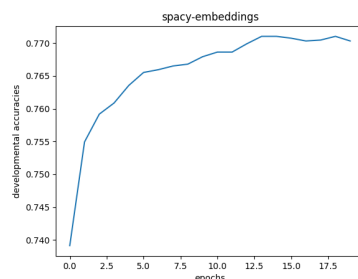
mean accuracy of best param: 0.7436521315614222

The accuracies generated by training with 10 epochs are higher. Therefore, I used the best hyperparameters from the first cross validation.

Training on the best hyperparameters

The best training accuracy is 77%. The test accuracy is 73.8%

The learning curve for training accuracy is



Plan until next milestone

- Evaluate how average perceptron performs on tfidf and roberta to determine which feature representation might be better
- Possibly experiment on the effects of the training set size