NLP Programming HW2

Jonas Nikula
20176392

1 Regularization tuning

First of I tried to tune the parameters of the logistic regressor. As I see it, the only parameter worth touching — at least for this exercise — is the C, or regularization parameter. However, changing it didn't have much of an effect, and the scikit-learn default, 1, seems to perform the best. Graph 1 depicts the effects of changing the parameter.

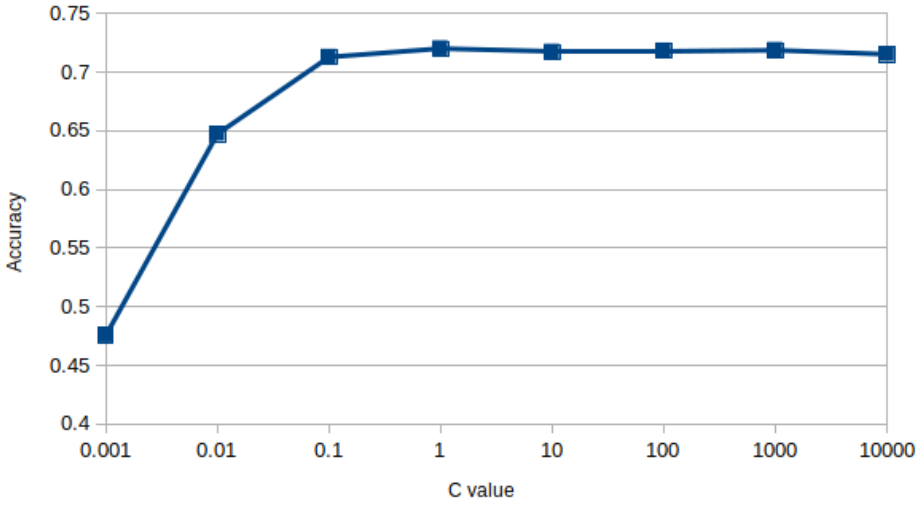

Figure 1: Effect of the regularization parameter on classifier accuracy

2 Tuning the n-grams parameter on word vectorization

Changing the n-gram range parameter on the word count vectorization had a bigger effect. The n-gram range option in this case means how many words are considered as a feature. The optimum value seemed to be a range of $1-2$. Table 1 has some different results.

Table 1: Some n-gram range values used in word count vectorization and the resulting accuracy

(1, 2)	0.7396
(1, 3)	0.7352
(1, 4)	0.7221
(2, 2)	0.7114
(2, 3)	0.6976
(2, 4)	0.6830
(3, 3)	0.6170
(3, 4)	0.5900
(4, 4)	0.4956

3 Trying out tf–idf

Perhaps somewhat surprisingly (at least to me), the results using term frequency were significantly worse than just using the word counts. The best score (obtained by (1,1) ngram range) was just 0.6973, which is worse than even the baseline.

4 Using the unlabeled data

The best results were gotten by adding some unlabeled data into the mix. The best accuracy on the development set is 0.7488, and that was achieved when adding 19,000 unlabeled samples to the training set.

The method for using the unlabeled data was the same as the first method described in the assignment description. To reiterate, we train the classifier on the labeled training set. Then we predict some labels for a part of the unlabeled set, I used a batch size of 100. Then we add that data into the "labeled" training set and repeat the process. In my implementation we just go through all the unlabeled data in this manner. As we can see from the graph 2, this works fairly well as the performance increases when add most of the unlabeled samples.

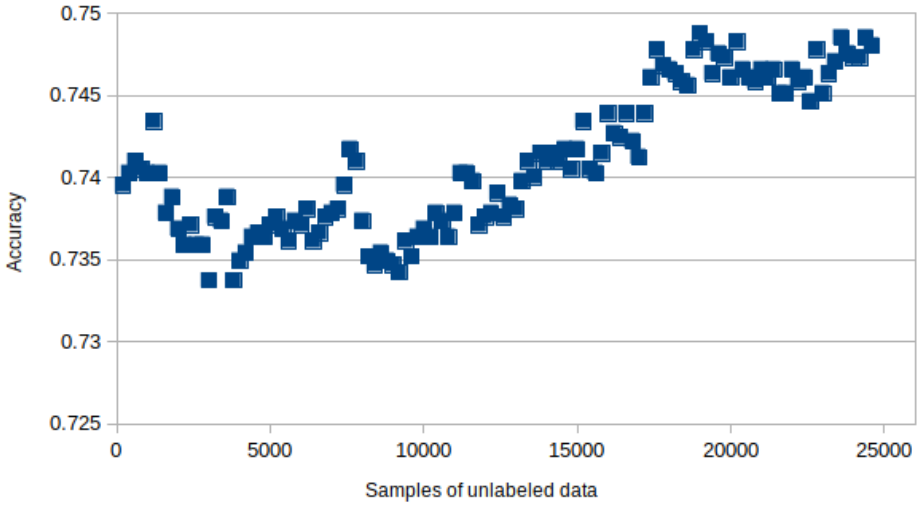

Figure 2: A XY-plot of the effect of adding unlabeled data samples to the training