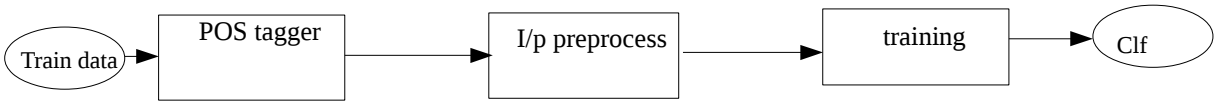
Design and implementation

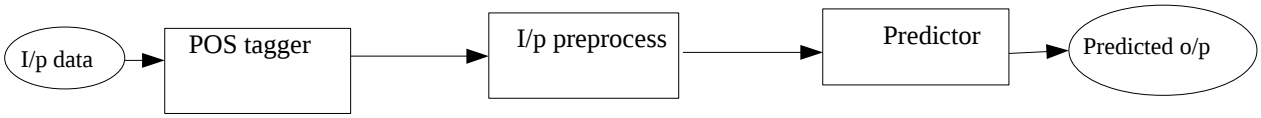
The sentiment analyzer is written in Python using scikit-learn toolkit.

The training and testing algorithm were designed as follows-

Training



Testing



Description of the sections

POS tagger

The Ark Twitter NLP tagger is used for the purpose of tokenizing the text and generating POS tags.

I/p preprocess

This step involves parsing output generated from the tagger into a form taken by the Predictor/Training stage. The main steps of process are -

- *Removing irrelevant words* – The tagger generated tags for urls/emails , determiners, usernames which are not relevant in context of sentiment analysis. These are filtered by this step.

Training

The preprocessed text is given frequency count , weighted by their TF-IDF weights.A multinomial Naive Bayes classifier is trained on the features thus formed. As an output the training algorithm stores a vocabulary and the classifier itself.

The classifier is trained using 10-fold cross validation on the training data and its mean accuracy came out to be ~75%. This later on, is replaced by SVM with linear kernel which gives a accuracy of ~77%

The classifier was trained via a 10-fold cross validation.Each fold was chosen such that the class ratio (0 /4) was maintained constant for each folds.

Predictor

The predictor uses the classifier stored in the training step to predict results.

Incremental Development

The only increments made to improve accuracy over the baseline is changing the base classifier from MultinomialNB to SVM, keeping the capitalized words etc. Changng parameters of the models etc.

Increase in accuracy is observed in

-Maintaining capitalization

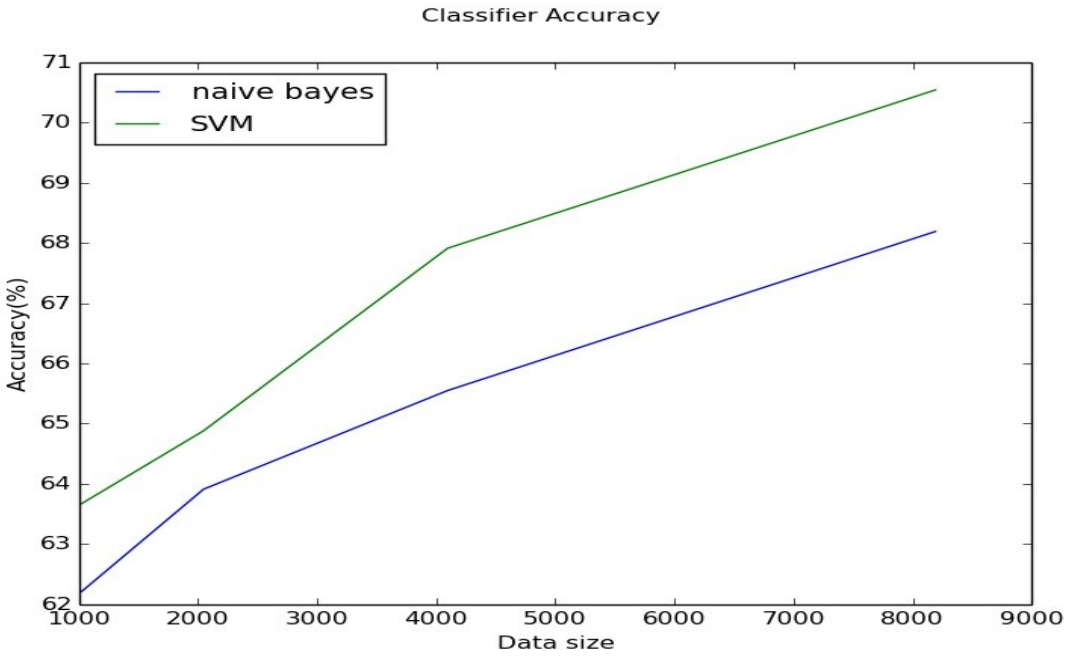
-Putting more weights to some words

-Removing stop words

Research

Some observations were made by taking training input sizes of 1024, 2048, 4096, 8192 finding the respective mean accuracies of the predicted.

Results are captured in the following graph-



The figure above clearly depicts the performance of two classifiers used in the assignment.

The green line indicates SVM which has higher accuracy than the naive bayes. Maximum accuracy obtained from Naive Bayes was ~68% compared to 70.5 % for SVM. Not only this, it can also be inferred the accuracy improves with increase in data size. The increase in training size causes accuracy to improve as the vocabulary size increases and the likelihood of various words increases.With larger data size (not plotted) higher accuracy results obtained.Training on input size of 0.9 of 1600000 gave 75.23% , 77.6% accuracies for Naive Bayes, SVM respectively.

References/ Acknowledgement

1. python docs and stackoverflow for implementation
2. class discussion at piazza
3. ark-NLP Twitter POS tagger
4. Speech and Language processing - Dan Jurafsky and Martin
5. http://sentiment.christopherpotts.net/