Siva Karthik Gade (sxg137530)

1. Introduction

This project is basically aimed at practicing the skills acquired through Natural Language Processing course, and have introductory experience to approaching a fully-fledged sentiment analysis task given a dataset. Have chosen ACML IMDB dataset for this purpose. Have preprocessed the data, trained different classifiers in a supervised manner on it, captured their accuracies of different preprocessing and classifier combinations, and also developed an ensemble classifier which employs a voting mechanism on individual classifiers' analysis predictions to classify test input. Voting mechanism is devised based on the performance of the individual classifiers on the dataset.

2. Problem Definition and Algorithm

2.1 Task Definition

In this project, have performed sentiment analysis task on ACML IMDB dataset. This dataset has over 50000 movie reviews. This dataset is processed through a few preprocessing tasks and classifiers are trained on it. Finally, an ensemble classifier is built using the individual classifier models. This project is a real world simulation of a supervised sentiment analysis task. ACML IMDB dataset is real world data, and working on it provided valued experience with the experimental science of NLP.

2.2 Algorithm Definition

There are three sub tasks involved in this task.

- **Data Preprocessing:** Have used out of the box StringToWordVector and AttributeSelection filters to tokenize the text and use only selected features.
- **Classifier Training:** Have trained Naïve Bayes, Decision Tree, Support Vector Machines (Degree1 and Degree2) and Random Forest classifiers on the dataset.
- **Ensemble Classifier Construction:** Constructed an ensemble classifier using the captured individual classifiers' models. Based on the performance of individual classifiers on the dataset, models included in the ensemble classifier and its voting mechanism are devised.

3. Experimental Evaluation

ACML IMDB data set has a total of 50000 files, split into positive and negative reviews. Due to computation power limitation, had to cut back on the size of the dataset to use for training and testing purposes.

- Have used a subset of dataset to model generation and capture classification accuracies of the individual models.
- Have used a different subset of the data to capture actual test performance of the ensemble classifier (consisting of best performing individual models) on the unseen examples.

The following are the three sub tasks involved in this project:

3.1 Data Preprocessing

3.1.1 StringToWordVector

This filter is used to convert all the text files into string tokens. Have modified a few of parameters' default values to suit the current task.

- TFTransform: Have experimented with and without considering term frequency. This is done to capture the effect of this preprocessing in sentiment analysis task.
- IDFTransform: Have experimented with and without considering inverse document frequency. Again, this is done to capture the effect of this preprocessing in sentiment analysis task.
- LowerCaseTokens: Made all the tokens case insensitive by lowering their case.
- MinTermFrequency: Have set minimum term frequency to 1.
- Stemmer: Have used the latest version of the snowball stemmer to bring the words back to their roots. Downloaded the latest source code, built it and used the jar file.
- Stopwords: Have used English stopwords. Note: Accuracies reported in below sections are captured with as well as without considering stopwords.
- Tokenizer: Have used NGram tokenizer for this task. Experimented with unigram and bigram tokens.
- WordsToKeep: Increased it to 1000000. This is done to ensure that all the tokens that satisfy above conditions are retained as features. Explicit feature selection is being handled by AttributeSelection classifier.

3.1.2 AttributeSelection

This filter is used to evaluate the importance of all the attributes and retain only those that satisfy the set influence cut off. Have used it with the following settings:

- Evaluator: Have used 'InfoGainAttributeEval' as the evaluator.

- Search: Have used 'Ranker' to rank attributes by their individual evaluations. Have set its threshold to 0 to reduce the feature list.

3.1.3 Summary

The above steps were performed as part of data preprocessing. The first step is performed, and then preprocessing and classification are done simultaneously. The application of both the filters is not done separately beforehand. It is done at the time of training classifiers itself. WEKA has provided FilteredClassifier to serve this purpose. It accepts filters (using MultiFilter) and classifier as its input. During generation of models, have used 3 fold cross validation and captured accuracies. During each fold, FilteredClassifier applies filters only on the training set of that fold, and tests the test set using the generated model. This tends to give more realistic predictions than preprocessing the complete training set (which includes test sets of different folds) before training the classifiers.

3.2 Classifier Training

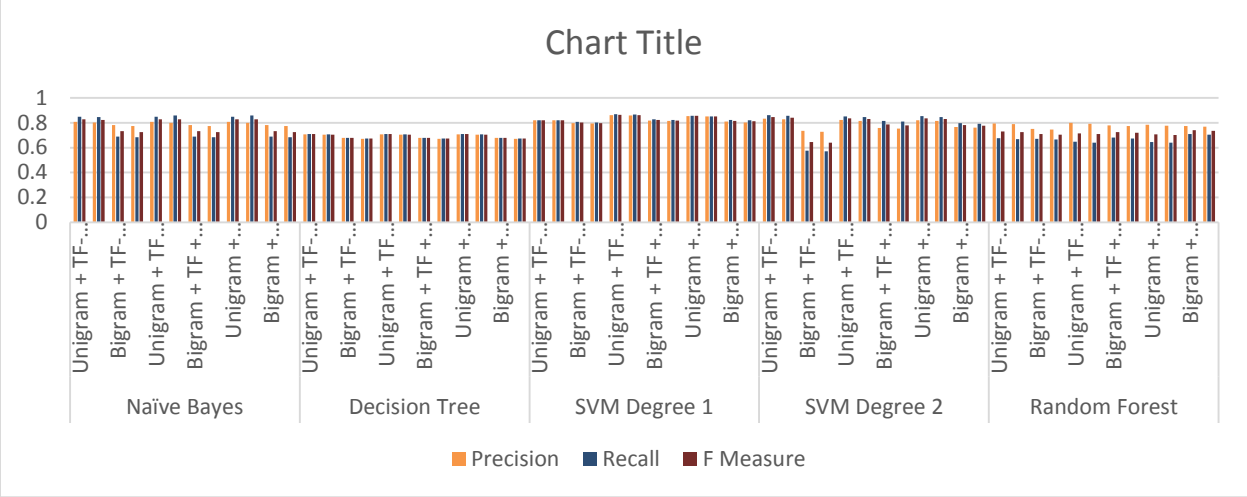
For this project, trained 5 different classifiers. Have used WEKA provided preprocessing and classifiers. Experimented with 12 different combinations of preprocessing for each of the classifiers. Listed below is the performance on sentiment analysis task by using different filter and classifier combinations.

DIFFERENT COMBINATIONS

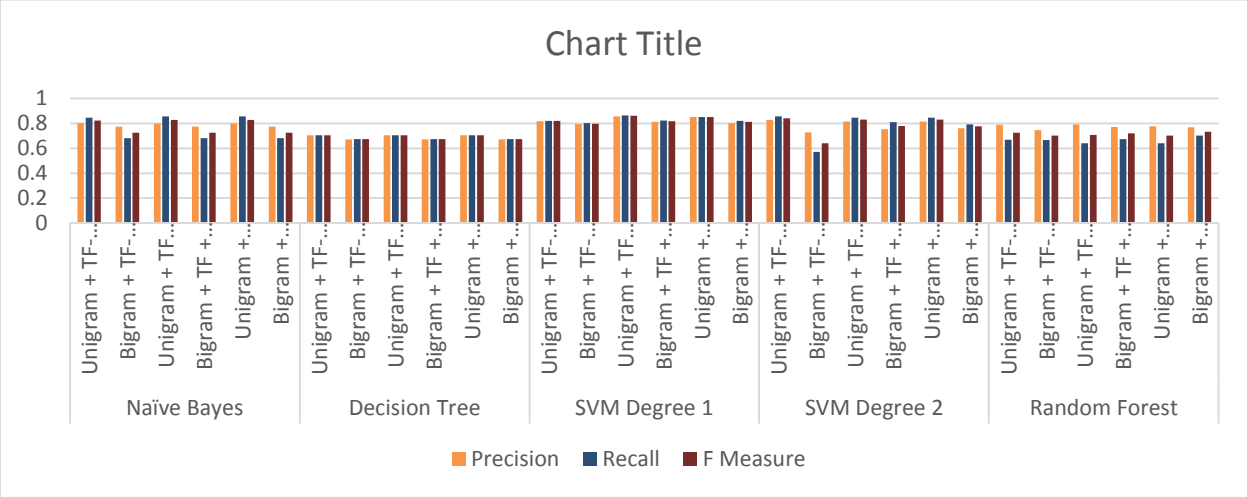
<table>
<tr><th>Classifier</th><th>Feature</th><th>Feature Value</th><th>Stop Words</th></tr>
<tr><td>Naïve Bayes</td><td rowspan="2">Unigram</td><td rowspan="2">TF – IDF</td><td rowspan="3">Exclude Stop Words</td></tr>
<tr><td>Decision Trees</td></tr>
<tr><td>SVM Degree 1</td><td rowspan="3">Bigram</td><td rowspan="2">TF</td></tr>
<tr><td>SVM Degree 2</td><td rowspan="2">Include Stop Words</td></tr>
<tr><td>Random Forests</td><td>Feature Present/Not</td></tr>
</table>

Classifier	Preprocessing Filters	TP	TN	FP	FN	Accuracy	Precision	Recall	F Measure
Naïve Bayes	Unigram + TF-IDF + Excluding Stopwords	2120	1990	510	380	82.2	0.806084	0.848	0.826510721
	Unigram + TF-IDF + Including Stopwords	2114	1977	523	386	81.82	0.801669	0.8456	0.823048472
	Bigram + TF-IDF + Excluding Stopwords	1724	2019	480	777	74.86	0.782214	0.689324	0.732837407
	Bigram + TF-IDF + Including Stopwords	1715	1987	501	797	74.04	0.773917	0.682723	0.725465313
	Unigram + TF + Excluding Stopwords	2120	1990	510	380	82.2	0.806084	0.848	0.826510721
	Unigram + TF + Including Stopwords	2143	1963	537	357	82.12	0.799627	0.8572	0.827413127
	Bigram + TF + Excluding Stopwords	1724	2019	480	777	74.86	0.782214	0.689324	0.732837407
	Bigram + TF + Including Stopwords	1715	1987	501	797	74.04	0.773917	0.682723	0.725465313
	Unigram + Feature P/NP + Excluding Stopwords	2120	1990	510	380	82.2	0.806084	0.848	0.826510721
	Unigram + Feature P/NP + Including Stopwords	2143	1963	537	357	82.12	0.799627	0.8572	0.827413127
	Bigram + Feature P/NP + Excluding Stopwords	1724	2019	480	777	74.86	0.782214	0.689324	0.732837407
	Bigram + Feature P/NP + Including Stopwords	1715	1987	501	797	74.04	0.773917	0.682723	0.725465313
Decision Tree	Unigram + TF-IDF + Excluding Stopwords	1775	1766	734	725	70.82	0.707453	0.71	0.708724296
	Unigram + TF-IDF + Including Stopwords	1763	1761	742	734	70.48	0.703792	0.706047	0.704918033
	Bigram + TF-IDF + Excluding Stopwords	1696	1694	806	804	67.8	0.677858	0.6784	0.678128749
	Bigram + TF-IDF + Including Stopwords	1680	1687	820	813	67.34	0.672	0.673887	0.672942119
	Unigram + TF + Excluding Stopwords	1775	1766	734	725	70.82	0.707453	0.71	0.708724296
	Unigram + TF + Including Stopwords	1763	1761	742	734	70.48	0.703792	0.706047	0.704918033
	Bigram + TF + Excluding Stopwords	1696	1694	806	804	67.8	0.677858	0.6784	0.678128749
	Bigram + TF + Including Stopwords	1680	1687	820	813	67.34	0.672	0.673887	0.672942119
	Unigram + Feature P/NP + Excluding Stopwords	1775	1766	734	725	70.82	0.707453	0.71	0.708724296
	Unigram + Feature P/NP + Including Stopwords	1763	1761	742	734	70.48	0.703792	0.706047	0.704918033
	Bigram + Feature P/NP + Excluding Stopwords	1696	1694	806	804	67.8	0.677858	0.6784	0.678128749
	Bigram + Feature P/NP + Including Stopwords	1680	1687	820	813	67.34	0.672	0.673887	0.672942119
SVM Degree 1	Unigram + TF-IDF + Excluding Stopwords	2052	2052	448	448	82.08	0.8208	0.8208	0.8208
	Unigram + TF-IDF + Including Stopwords	2047	2049	453	451	81.92	0.8188	0.819456	0.819127651
	Bigram + TF-IDF + Excluding Stopwords	2016	1986	514	484	80.04	0.796838	0.8064	0.801590457
	Bigram + TF-IDF + Including Stopwords	2003	1981	521	495	79.68	0.793582	0.801841	0.797690163
	Unigram + TF + Excluding Stopwords	2172	2146	354	328	86.36	0.859857	0.8688	0.864305611
	Unigram + TF + Including Stopwords	2162	2141	359	337	86.07722	0.857596	0.865146	0.861354582
	Bigram + TF + Excluding Stopwords	2072	2035	465	428	82.14	0.816713	0.8288	0.822711932
	Bigram + TF + Including Stopwords	2059	2028	471	442	81.74	0.813834	0.823271	0.818525144
	Unigram + Feature P/NP + Excluding Stopwords	2141	2132	369	358	85.46	0.852988	0.856743	0.85486125
	Unigram + Feature P/NP + Including Stopwords	2133	2119	373	375	85.04	0.851157	0.850478	0.85081771
	Bigram + Feature P/NP + Excluding Stopwords	2054	2013	487	446	81.34	0.808343	0.8216	0.814917675
	Bigram + Feature P/NP + Including Stopwords	2042	2009	502	447	81.02	0.802673	0.82041	0.811444467
SVM Degree 2	Unigram + TF-IDF + Excluding Stopwords	2152	2069	431	348	84.42	0.83314	0.8608	0.846744049
	Unigram + TF-IDF + Including Stopwords	2138	2055	447	360	83.86	0.827079	0.855885	0.841235491
	Bigram + TF-IDF + Excluding Stopwords	1439	1984	516	1061	68.46	0.736061	0.5756	0.646015713
	Bigram + TF-IDF + Including Stopwords	1423	1971	532	1074	67.88	0.727877	0.569884	0.639263252
	Unigram + TF + Excluding Stopwords	2129	2037	463	371	83.32	0.821373	0.8516	0.836213668
	Unigram + TF + Including Stopwords	2112	2024	480	384	82.72	0.814815	0.846154	0.830188679
	Bigram + TF + Excluding Stopwords	2039	1854	646	461	77.86	0.759404	0.8156	0.786499518
	Bigram + TF + Including Stopwords	2015	1848	662	475	77.26	0.752708	0.809237	0.779949681
	Unigram + Feature P/NP + Excluding Stopwords	2130	2034	466	370	83.28	0.820493	0.852	0.835949765
	Unigram + Feature P/NP + Including Stopwords	2113	2024	478	385	82.74	0.815515	0.845877	0.83041855
	Bigram + Feature P/NP + Excluding Stopwords	1993	1893	607	507	77.72	0.766538	0.7972	0.781568627
	Bigram + Feature P/NP + Including Stopwords	1981	1879	622	518	77.2	0.761045	0.792717	0.776558212
Random Forest	Unigram + TF-IDF + Excluding Stopwords	1689	2065	435	811	75.08	0.795198	0.6756	0.730536332
	Unigram + TF-IDF + Including Stopwords	1673	2054	447	826	74.54	0.789151	0.669468	0.724399221
	Bigram + TF-IDF + Excluding Stopwords	1676	1946	554	824	72.44	0.75157	0.6704	0.708668076
	Bigram + TF-IDF + Including Stopwords	1662	1936	568	834	71.96	0.745291	0.665865	0.703343208
	Unigram + TF + Excluding Stopwords	1620	2092	408	880	74.24	0.798817	0.648	0.715547703
	Unigram + TF + Including Stopwords	1601	2079	423	897	73.6	0.791008	0.640913	0.708093764
	Bigram + TF + Excluding Stopwords	1700	2014	486	800	74.28	0.777676	0.68	0.725565514
	Bigram + TF + Including Stopwords	1687	1998	496	819	73.7	0.77279	0.673184	0.719556409
	Unigram + Feature P/NP + Excluding Stopwords	1614	2052	448	886	73.32	0.782735	0.6456	0.707584393
	Unigram + Feature P/NP + Including Stopwords	1603	2040	460	897	72.86	0.777024	0.6412	0.702607933
	Bigram + Feature P/NP + Excluding Stopwords	1776	1981	519	724	75.14	0.773856	0.7104	0.740771637
	Bigram + Feature P/NP + Including Stopwords	1763	1960	533	744	74.46	0.767857	0.703231	0.734124506

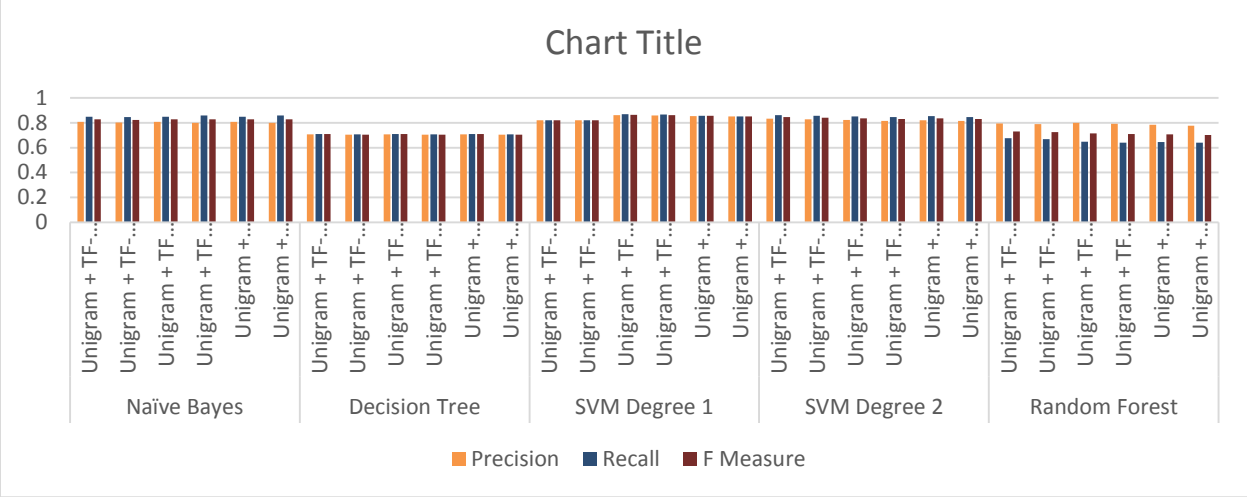
Excluding Stop Words



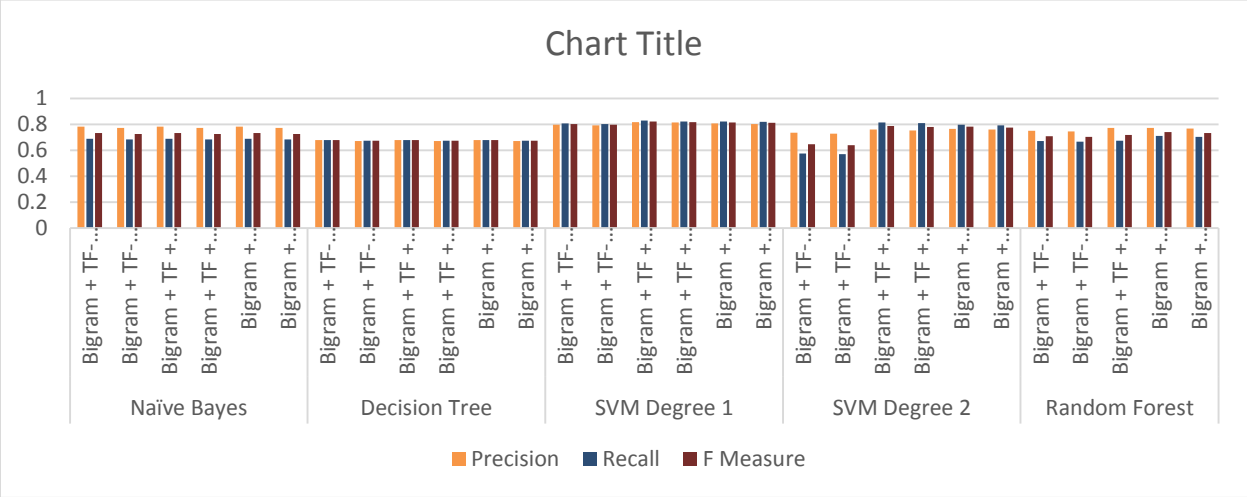
Including Stop Words



Unigrams



Bigrams



Performance Analysis

- Above listed are the accuracies recorded by all the individually learnt classifier models.
- The data is linearly separable. Linear classifiers recorded the highest accuracies.
- Classification accuracies descending order: SVM Degree 1, SVM Degree 2, Naïve Bayes, followed by Decision Trees and Random Forest.
- The accuracies recorded while excluding stop words tend to be slightly better than the accuracies recorded while including them.
- The accuracies recorded with unigram token models are better than their respective bigram counter parts. Only in case of Random Forests, better accuracies are observed with bigram tokenization.
- The accuracies recorded while using TF-IDF tended to be slightly higher than using TF or feature present/not present metric. But in most cases, they didn't differ much across this variable changing.
- Over all, the classifiers and filter combinations chosen worked well on this movie review sentiment analysis task. Most of the accuracies recorded are in higher 70s and lower 80s. Corresponding precision, recall and F-measures also in agreement.

All the scripts, code and datasets used references are available in the submitted artifacts archive.

3.3 Ensemble Classifier for Sentiment Analysis

Using the results of the experiments done above, have constructed an ensemble classifier to perform Sentiment Analysis on the test data. This classifier uses the predictions made by the best individual classifier models, employs a voting mechanism on those results and predicts the class of the test input. The below subsections describe the individual classifiers' models generation, details of the voting mechanism employed in this ensemble classifier to predict the class of the test input.

3.3.1 Training Models

The ensemble classifier needs trained models of the individual classifiers (Naïve Bayes, Decision Trees, SVM Degree1, SVM Degree2 and Random Forest) in order to make a decision using voting. Among the 12 combinations of filters & preprocessing experimented with each classifier, have chosen the best model of each classifier to construct this ensemble classifier.

3.3.2 Voting Mechanism

For any test input, all the 5 chosen individual classifiers output their predictions. Predictions made by these are used in the following way to come to a final prediction:

- If there is a strict majority of a particular prediction, then that is output as the final prediction of the ensemble classifier.
- If above case fails to determine output, then weighted sum of votes is taken into account. In this process, output of Naïve Bayes and SVM (degree 1 kernel and degree 2 kernel) are given a weightage of 2 each. Output of remaining classifiers (Decision Tree and Random Forest) are given weightage of 1 each. The prediction with the most weighted sum is output as the final prediction of the ensemble classifier.
- If above two mechanisms fail to provide a result, then output of SVM (degree 1 kernel) classifier model is output as the final prediction of the ensemble classifier.

Code of this ensemble classifier is available in the artifacts archive submitted.

3.3.3 Accuracy

Have tested this ensemble classifier on a subset (1500 files) of the unseen ACML IMDB test dataset. This classifier recorded accuracy of **90.4% (Precision: 0.896, Recall: 0.913, F-Measure: 0.904)** on that test set, which is far better than the accuracy reported by any of the individual classifiers.

4. Future Work

Due to limited computation resources, couldn't try the below variations. Would have preferred performing them as well which would lead to better evaluation of the dataset and better sentiment analysis for testing on unseen reviews.

- Train models on the complete dataset.
- Try higher order n-grams and some novel semantic feature selection heuristics.
- Try different other classifiers as well which were taught through this course (Maximum Entropy ...).

5. Conclusion

ACL IMDB is a very good dataset. Linear SVM tend to perform best on this data set with including stop words. Through this project, able to gain better understanding of performing the sentiment analysis task and text preprocessing steps that needed to be carried out in order to gain better relevant results through classification. Have used TextDirectoryToARFF, StringToWordVector, AttributeSelection, J48, NaiveBayes, SMO, IbK, RandomForest provided in WEKA. The ensemble classifier developed should work very well even in real time on unseen samples, as supported by the earlier mentioned testing results on unseen examples.

6. Bibliography

http://www.hlt.utdallas.edu/~yangl/cs6320/

http://www.cs.waikato.ac.nz/ml/weka/

http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz

http://www.esp.uem.es/jmgomez/tmweka/index.html

https://github.com/jmgomezh/tmweka/blob/master/FilteredClassifier/MyFilteredClassifier.java