Project 2
Twitter Sentiment Classification Report

Team Members
Sreejith Menon (smenon8@uic.edu)
Rajan Bhandari (rbhand4@uic.edu)

Table of Contents

Abstract

Sentiment Classification problem in general involves analyzing text in a document or sentence to classify its polarity i.e. whether the opinion expressed in the document or sentence is positive, negative or neutral. Similarly, for our analysis we were given 2 labeled training datasets – tweets gathered for 'Obama' and 'Romney' during their presidential contest and each tweet in the datasets was labeled as +1 (for positive opinion) or -1 (for negative opinion) or 0 (for neutral opinion). Our task was to analyze the training data for Obama and Romney separately and build classifiers that would classify the test data for Obama and Romney into +1, -1 or 0 classes. We started with data pre-processing steps involving data cleaning (removal of html tags, links and special characters), stop words removal and stemming and the processed data was then used to generate a vocabulary (set of features). Using the set of features we transformed tweets into bag of words model and represented each tweet using tf-idf values for all features. We trained 4 classifiers – Naïve Bayes, Logistic Regression, Support Vector Machines and Ada Boost and performed a 10-fold cross validation initially, which showed that Logistic Regression classifier performed better than other classifier for the given training data. Our results were further confirmed by running these classifiers on the actual test data with Logistic Regression providing higher overall accuracy and F1-score for positive and negative class.

Introduction

The sentiment classification for ‘Romney’ and ‘Obama’ problem involved creating classifiers based on supervised learning techniques as we had labeled training datasets. Our initial analysis of the training data revealed that Romney data was biased as it had more than 50% of tweets classified as ‘Positive’ (Figure 1).

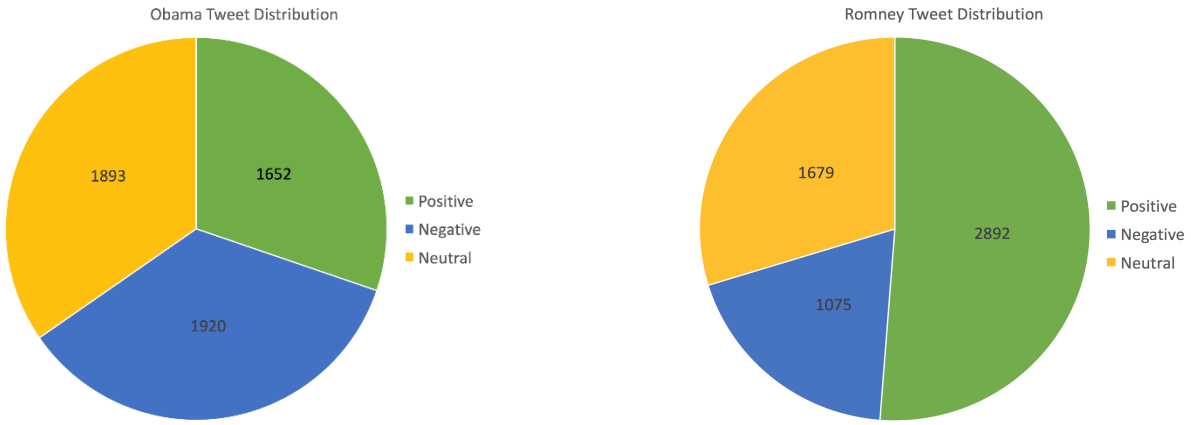

Figure 1

The initial analysis also revealed presence of lots of html tags, hyperlinks and special characters with no sentimental values and confirmed the steps that needed to be performed during data pre-processing phase. To reduce the number of features to be used in bag of words model, we also used techniques like stop words removal and stemming using standard ‘nltk’ library in Python during data pre-processing phase.

As you can see from Figure 1, the class distributions for Obama seem equal while for Romney it seems that the majority of the tweets we have are negative. Any classifier that will be trained directly on the dataset will be slightly biased towards the negative class.

Techniques

Data Pre-processing

Based on initial analysis of data below pre-processing steps were performed:

1. **Data Cleaning**
   The tweets initially contained lots of html tags, hyperlinks and other special characters which had to be cleaned as they contributed no sentimental value towards classification. After removal of special characters, we converted the words in tweets to lowercase which also helped in feature reduction by eliminating distinction between same words in uppercase and lowercase format.

   Example: -
   Obama tweet before data cleaning:
   Not all of Hollywood has his back! RT @RedAlert: Gene Simmons Yanks <e>Obama</e> Support, Calls Him a “Piss-Poor” President http://t.co/joKcRcDF

   Obama tweet after data cleaning (converted into list of features):
   ['not', 'all', 'of', 'hollywood', 'has', 'his', 'back', 'rt', 'redalert', 'gene', 'simmons', 'yanks', 'obama', 'support', 'calls', 'him', 'a', 'pisspoor', 'president']

2. **Stop words removal**
   In order to reduce the number of features we also removed common occurring stop words in English using the Natural Language Translation Kit (nltk) library provided in Python.

   Example: -
   List of a tweet’s features before stop words removal:
   ```
   ['not', 'all', 'of', 'hollywood', 'has', 'his', 'back', 'rt', 'redalert', 'gene', 'simmons', 'yanks', 'obama', 'support', 'calls', 'him', 'a', 'pisspoor', 'president']
   ```

   List of a tweet’s features after stop words removal:
   ['hollywood', 'back', 'rt', 'redalert', 'gene', 'simmons', 'yanks', 'obama', 'support', 'calls', 'pisspoor', 'president']

3. **Stemming**
   We also used stemming to reduce the number of features by reducing words in tweets to their original form. The words in tweets were stemmed using the PorterStemmer module provided in Natural Language Translation Kit (nltk) library provided in Python.

   Example: -
   List of a tweet’s features after cleaning and stop words removal.
   ```
   ['hollivan', 'hereistheanswer', 'youre', 'missing', 'point', 'im', 'afraid', 'understand', 'bigger', 'picture', 'dont', 'care', 'obama', 'elected']
   ```

List of a tweet's features after stemming.

```
['hollivan', 'hereistheansw', 'your', 'miss', 'point', 'im', 'afraid',
'understand', 'bigger', 'pictur', 'dont', 'care', 'obama', 'elect']
```

Feature Extraction

After the completion of data pre-processing steps on the training data, we generated a vocabulary (set of features) to convert the tweets into bag of words model. We represented each tweet as a vector of tf-idf values of all features.

Example (for clarification only as the actual vocabulary is very large):

If vocabulary contains below features

```
['hello','obama','win','to', 'result','bad']
```

And, a tweet's features are as below

```
['win','to','obama']
```

Then, vector representation of the tweet (dummy tf-idf values represented in example below)

```
[0,0.3,0.4,0.6,0,0]
```

Classification Methods

We tried 4 different classifiers viz. Naïve Bayes, Support Vector Machines, Logistic Regression and Ada-Boost, the results of their 10-fold cross fold validation for the training data provided are as shown below. For the training data, Logistic Regression performed the best(on an average) for both Obama and Romney datasets as confirmed by its F1-score.

For Obama:

Classifier	Overall Accuracy	Positive Class			Negative Class			Neutral Class		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Naïve Bayes	0.555	0.519	0.699	0.595	0.61	0.576	0.592	0.543	0.409	0.467
SVM	0.562	0.624	0.506	0.560	0.556	0.667	0.607	0.522	0.505	0.513
Logistic Regression	0.58	0.626	0.539	0.579	0.592	0.652	0.621	0.533	0.543	0.538
AdaBoost	0.490	0.535	0.512	0.523	0.487	0.596	0.536	0.447	0.362	0.4

For Romney:

Classifier	Overall Accuracy	Positive Class			Negative Class			Neutral Class		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Naïve Bayes	0.543	0.522	0.163	0.249	0.569	0.865	0.687	0.424	0.231	0.299
SVM	0.536	0.630	0.098	0.170	0.536	0.968	0.690	0.484	0.072	0.126
Logistic Regression	0.563	0.594	0.228	0.330	0.584	0.883	0.703	0.443	0.226	0.299
AdaBoost	0.535	0.449	0.225	0.3	0.557	0.885	0.684	0.430	0.133	0.203

Evaluation

Our classification results for the final test data are listed below where Logistic Regression again performed better than other classifiers for both Obama and Romney datasets as shown by its high F1-score.

For Obama:

Classifier	Overall Accuracy	Positive Class			Negative Class			Neutral Class		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Naïve Bayes	0.544	0.513	0.615	0.559	0.576	0.578	0.577	0.544	0.45	0.493
SVM	0.539	0.561	0.5	0.529	0.545	0.629	0.584	0.514	0.483	0.498
Logistic Regression	0.554	0.581	0.506	0.541	0.570	0.614	0.591	0.517	0.533	0.524
AdaBoost	0.486	0.451	0.601	0.515	0.510	0.577	0.541	0.506	0.296	0.374

For Romney:

Classifier	Overall Accuracy	Positive Class			Negative Class			Neutral Class		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Naïve Bayes	0.567	0.659	0.166	0.265	0.567	0.921	0.702	0.528	0.232	0.322
SVM	0.555	0.644	0.174	0.274	0.547	0.969	0.699	0.610	0.104	0.178
Logistic Regression	0.61	0.695	0.361	0.475	0.618	0.886	0.728	0.526	0.306	0.387
AdaBoost	0.554	0.490	0.327	0.392	0.569	0.866	0.687	0.524	0.172	0.260

Conclusion

Logistic regression seems to be the classification method of choice in our case. The accuracies and the performance metric seems to hit a ceiling around 60% for the Obama dataset and 65% for the Romney dataset. We can infer that this is the best possible metric that can be achieved using only classification techniques. Use of bi-grams and tri-grams could improve the accuracies slightly but no significantly. The next logical step for improvement would be to step on to the natural language processing arena, where we can use advanced techniques to capture the context of words and then train an intelligent classifier that can take into account context information.