Twitter Sentiment Analysis

Md Raihan Majumder
Department of Computer Science
Texas Tech University

Sraddhanjali Acharya
Department of Computer Science
Texas Tech University

***Abstract*—We implemented an approach for classifying tweets based on their sentiments. Sentiments are divided into positive and negative emotions. This research is very useful for consumers how want to buy any product or for companies who want to know how their products are doing in the market. We used Naive Bayes, Maximum Entropy, and SVM for the classification to observe their performance.**

1. Introduction

Twitter is a popular microblogging website where people can create characterbound (max 140 characters) short messages called Tweets. These tweets allow users to give their expression on different topics ranging from politics, sports, entertainment, social affairs etc. We attempt to build a twitter sentiment to classify tweets as Positive, Negative based on their sentiment. Twitter sentiment analysis is important because it can provide mass peoples polarity towards a topic. Political party may know whether people support their agenda or not. A company may know how their products are doing. Consumer can learn about a product before making a final decision purchasing it. Twitter sentiment analysis is different from those of movie review analysis because movie reviews are more thoughtful and well written. But, tweets are usually very casual and light in manner. Yet, it gives value information to companies as a form of feedback.

In order to classify tweets, we used different machine learning classifiers such as Support Vector Machines, Naive Bayes Classifier, Maximum Entropy Classifier. As all the classifiers, we used, are supervised learning technique, we need training data. For our purpose, we used a publicly available dataset called *2008 US Election debate by Nick Diakopoulos and Shamma, D.A.*

1.1. Related Work

Alec et. al [1] classified tweets as positive or negative using an unweighted unigram model and the tweets were labelled using distance supervision. Unlike [1] which removed emoticons regarding them as stopwords, Read [2] identified the process and differentiated emoticons of various sentiments as positive and negative features in a weighted feature model for tweet dataset. A. Pak et. al [4] performed a linguistic inspection to build classifier to classify documents as positive, negative or neutral. In addition, they replace each emoticons with their respective sentiments similar to [3]. Pang and Lee [3] investigated the performance of various classifiers such as Naive Bayes, Support Vector Machines and Maximum Entropy in classifying the sentiments of users' movie reviews with accuracies of 81.0%, 82.9% and 80.4%. The paper removed the emoticons which could have served as feature to obtain the sentiment of the tweets. L. Jiang et al [5] focuses on target dependent classification, in which a query is given first, based on which tweets are positive, negative or neutrall based on the sentiment of target query. The paper proposes improvements for targetdependent Twitter sentiment classification by integrating target-dependent features, and considering related tweets while classifying.

2. Baseline

In this section, the description of classifiers, the evaluation of classifiers and the performance measures are discussed. The classification of the twitter dataset is a supervised learning since we are equipped with labelled dataset. Classification comprises of training and testing phase of the dataset. Each input value of the training set are converted into a feature value in the feature set. The feature sets should capture the information of the inputs.

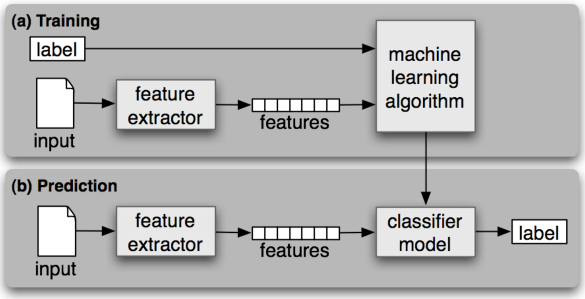

Figure 1. Example of basics of Machine Learning Algorithms

The feature vector and labels are fed into the machine learning algorithm to obtain a model for the classifier. In the testing phase, i.e the prediction phase, the feature vector built earlier will be used to convert unseen inputs to feature sets. The feature set is then fed into the model and predicted labels are obtained. We compare if the model correctly classified this unseen input or not, in the performance measures. A general illustration is given in the Figure 1 [6].

2.1. Naive Bayes Classifier

A Naive Bayes Classifier is a classifier which predicts a class value given a set of set of attributes. For each known class value,

- Calculate probabilities for each attribute, conditional on the class value.
- Use the product rule to obtain a joint conditional probability for the attributes.
- Use Bayes rule to derive conditional probabilities for the class variable.
- Once this has been done for all class values,

The class with the highest probability is considered as the probable class.

Naive Bayes Classifier assumes that all the features are unrelated to each other. Presence and absence of a feature does not influence the presence or absence of any other feature. Naive Bayes can be used for Binary or Multiclass classification.

2.2. Maximum Entropy Classifier

The Maximum Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. Unlike the Naive Bayes classifier that we discussed in the section 2.1, the maximum entropy classifier does not assume anything before about features. Imagine 3 buckets with 10 red and 10 blue balls as shown in the Figure 2. The first and last have well separated (or ordered) blue and red balls, and have low entropy. The middle one has uniformly placed red and blue balls (unordered or random) and has high entropy. So, maximum entropy is achieved when we have uniform distribution of things or in other words when they have the most randomness. Maximum Entropy classifier prefers the uniformity or maximum entropy if no data is observed. But as it sees the data, it has to move away from the maximum entropy by explaining data. After it has explained the data, it again tries to maximize the entropy on whatever remaining is not seen. The Max Entropy classifier can be used to solve a large variety of text classification problems such as language detection, topic classification, sentiment analysis and more.

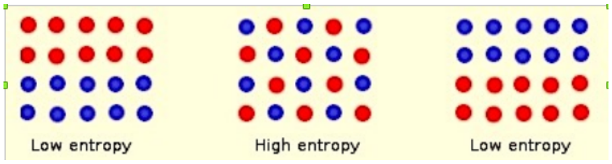

Figure 2. Maximum Entropy Demonstration

2.3. Support Vector Machine

Support Vector Machine (SVM) is a popular classification method that performs classification tasks by constructing hyperplanes in a multidimensional space. The goal of support vector machine is to find an optimal separating hyperplane which separates two classes best where best means biggest margin between classes.Even if we find different separating hyperplanes, it may not be optimal. A best hyperplane is one that is as far as possible from data points from each category. Once we get the hyperplane, we calculate the distance between hyperplane to the closest datapoint. Once we double it, we get margin. The optimal hyperplane will be the one with the biggest margin as shown in Figure 3.

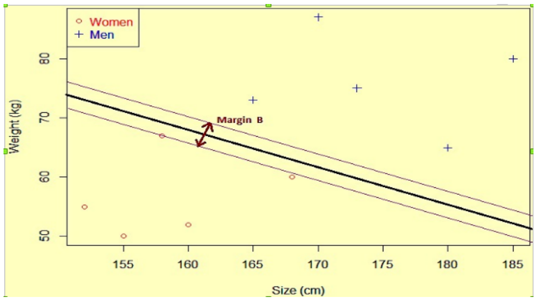

Figure 3. Best Hyperplane with Margin B

2.4. Cross Validation

Cross Validation is a validation method to check how generalized the statistical analysis is to an independent dataset. The main purpose of crossvalidation is to ensure that, we do not lose significant amount of data, which could either help in modeling or testing, when we pick our partitions, like in for example a traditional validation method uses 70% partition of dataset for training and 30% for testing, which can be a problem. This is avoided in cross validation because it averages all the prediction errors and helps to derive a general estimate which is generalized over new datasets as well, which can give more accurate results. The two types of crossvalidation techniques are as follows:

1) **Exhaustive cross validation:** The sample is exhaustively divided into all possible partitions of training and testing dataset.
2) **Nonexhaustive cross validation:** The sample is not exhaustively divided into all possible partitions of training and testing dataset.

In *k*fold cross validation, the dataset is divided randomly into k partitions of equal size. One of the k sample is used as test dataset, and the remaining *k 1* are used as the training dataset. The cross validation process is done k times, with different sample as test and *k 1* as training dataset. The results from these k folds are averaged to give a single values. K fold cross validation ensures the observations from the dataset are use for validation only once. K fold stratified cross validation is same as kfold cross validation, with the difference that the kfold have equal number of samples from both classes.

However, there are some limitations and issues with the cross validation techniques, such as the limitation that, for a cross validation to produce useful result, the training and the testing dataset need to be from the same population. In scenarios where the nature of the dataset evolves over time, the model becomes unrealistic as the new dataset evolve from the old nature over which the model was prepared. This can wrongly conclude that, the model can not classify certain datasets, when the case maybe the model is not generalized enough.

2.5. Performance measures

The measures of performance of machine learning algorithms are done by various metrics such as precision, accuracy, F, recall. etc.

1) **Accuracy** is the percentage of total correct predictions over the total number of predictions made. Accuracy measure alone can be misleading since if the dataset has a large class imbalance, the model might have a high classification accuracy due to the majority class.
2) **Confusion matrix** can be used a way to present the classification results in a table.

Table 1. Confusion Matrix for Two Sentiments

Sentiments	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Each cell on the table contains the number of predictions made by the classifier. If the classifier is perfect, all the predictions will fall under True Positive and True Negative. However, if incorrect predictions are made, the cells False Negative and False Positives will be populated.

3) **Precision** is the total number of True Positive over the total of True Positives and True Negatives i.e $TP/(TP + TN)$. A high value of precision means a large number of True Positives. It is a measure of classifier exactness.
4) **Recall** is the number of True Positive over the total of True Positives and False Negatives i.e $TP/(TP + FN)$. It is the measure of classifier's completeness.
5) **F1 Score** is calculated as $2 * ((precision * recall)/(precision + recall))$ and is a measure of balance between the recall and precision measures.

3. Problem Definition

3.1. Characteristics of Tweets and Preprocess of Tweets

Tweets have informal language constructs with emoticons such as : P, :) and lots of characters like @, #, and URl links. Leaving the emoticons in can have negative effect on the accuracies of Max Entropy and SVM classifiers but little in Naive Bayes [3].

1) usernames with @ symbol (e.g @twitter) can be replaced with a fixed AT_USER token.
2) hashtags with # symbol (e.g #tweetdebate) can be replaced with the word "tweetdebate" token.
3) multiple whitespaces are removed with just a single whitespace.
4) punctuations in the tweets (e.g Cool! replaced wtih Cool) are removed with just the word.
5) urls with "/", "//" (e.g http://machinelearning.org) can be replaced wtih just URL token.
6) stop words such as "as", "is", "and", "the", "an" can be removed from the tweets.
7) repetition of letters (e.g coooool, chillllll) is prevalent in tweets becaue of the informal setting.

3.2. Building Feature Vector

The filters mentioned in above section are used beforehand. In the unweighted unigram model, the presence and abscence of words that appear in tweet are taken as features. The addition of single words into features intuitively calls this approach a "unigram" approach.

3.3. Experimental Evaluation

We used the dataset *2008 US Election debate by Nick Diakopoulos and Shamma, D.A*. In the first phase of the project, the dataset has a tab separated data values of tweetid, date of tweet generation, tweet, author of the tweet, ratings1, ratings2, ratings3, ratings4, ratings5, ratings6, and ratings7. The 7 ratings here are labels 1, 2, 3, 4 given to the tweet. The labels used were Negative(1), Positive(2), Mixed(3), Other(4).

1) A majority voting on the 7 ratings to assign labels to the tweets of positive, negative and neutral and the mixed and other samples were merged into Neutral label. Then, the number of positive, negative, neutral tweets were 774, 1309 and 1196 respectively.
2) The tweets were preprocessed as explained in section 3.1.
3) The tweets were processed to generate feature list. Unigram feature extractor model extracts feature based on if a tweet contains a feature or not. If the feature is present, then value in the feature list is 1 else it is 0.

4) The traditional traintest split of 90 and 10 was done.
5) The SVM classifier was trained using libsvm library with a linear kernel, RBF kernel and POLY kernel with various values of C and Gamma.
6) The classification accuracy of around 39.64% was obtained for values C=100, degree=3, gamma=0 for SVM with POLY kernel. We deemed this result as failure due to unbalanced dataset of the classes and the merging of mixed and other into a single label (neutral).
7) Instead, we extracted an equal number of positive and negative labelled tweets (744) from the debate dataset.
8) The tweets were preprocessed before obtaining the filtered feature lists as explained in section 3.1.
9) Python Natural Toolkit Library was used to obtain the feature vectors.
10) The dataset was split into 75 25 train and test for 1fold and 5fold cross validation purpose.
11) The crossvalidation was run on SVM, Maximum Entropy Classifier, and Naive Bayes Classifier using the nltk library.
12) We obtained 80% accuracy for Naive Bayes Classifier, 79% for Maximum Entropy Classifier and 78% for SVM in five fold cross validation.

4. Results and Discussions

The performance metrics for the debate dataset is in Table 2.

Table 2. PERFORMANCE METRICS FOR DEBATE DATASET

Measures	Accuracy	Precision	Recall	F Measure
Classifiers	Single, Five	Single, Five	Single, Five	Single, Five
SVM	63, 78	63, 78	63, 78	63, 78
Max Ent	68, 79	70, 80	68, 79	67, 79
Naive	69, 80	70, 81	69, 80	69, 80

We can observe that for classifier SVM, the values of precision, recall are equal. It means, the algorithm classified equal amount of tweets as false positives as it classified false negatives. We deemed this result to be equal because of the equal number of each classes of i.e. positive and negative labelled tweets.

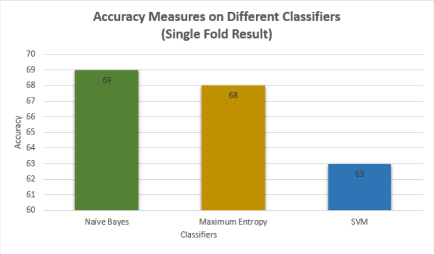

Figure 4. Accuracy Measure on Different Classifiers Single Fold Result

In Figure 4, for the performance measure accuracy, the Naive Bayes classifier outperforms MaxEntropy classifier and SVM in classifying the tweets correctly with 69% in single fold cross validation. However, the outperformance is not substantial since the other classifiers are also within similar range around 69%.

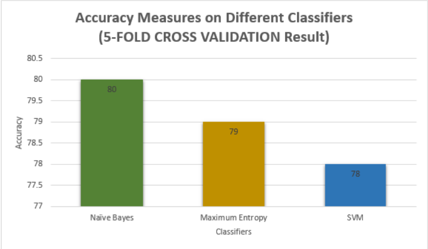

Figure 5. Accuracy Measure on Different Classifiers Five Fold Result

In Figure 5, in accuracy measure, the Naive Bayes classifier outperforms MaxEntropy classifier and SVM in classifying the tweets correctly with 80% in five fold cross validation.

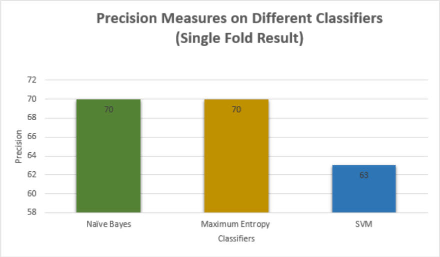

Figure 6. Precision Measure on Different Classifiers Single Fold Result

In Figure 6, Naive Bayes and Maximum Entropy classifier have 70% precision whereas SVM has 63% in the single fold cross validation.

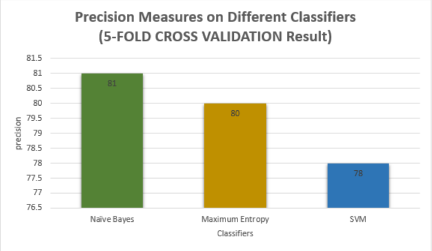

Figure 7. Precision Measure on Different Classifiers Five Fold Result

In Figure 7, after doing a five fold cross validation, Naive bayes slighly outperformed the rest of the classifiers with precision value of 81%.

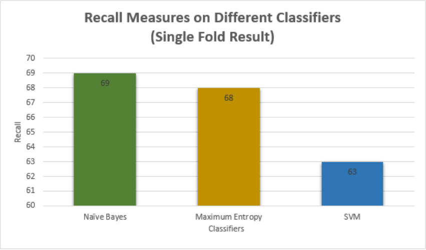

Figure 8. Recall Measure on Different Classifiers Single Fold Result

In Figure 8, the recall value for Naive Bayes is 69%, 68% for Max Entropy and 63% for SVM for single fold cross validation.

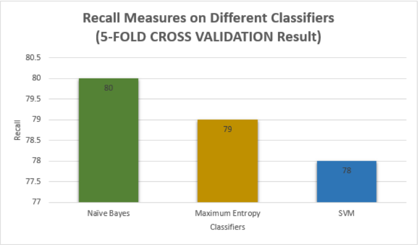

Figure 9. Recall Measure on Different Classifiers Five Fold Result

In Figure 9, the recall value for Naive Bayes is 80%, 79% for Max Entropy and 78% for SVM for five fold cross validation.

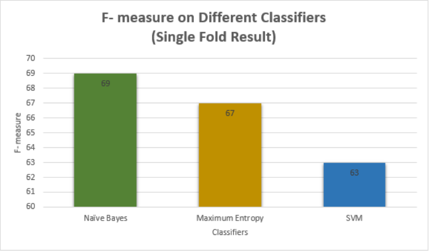

Figure 10. F Measure on Different Classifiers Single Fold Result

In Figure 10, the f measure for Naive Bayes is 69%, 67% for Max Entropy and 63% for SVM for single fold.

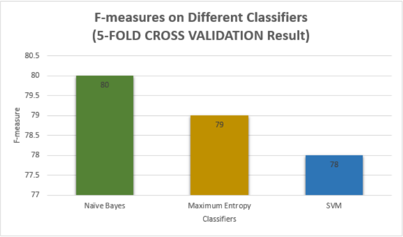

Figure 11. F Measure on Different Classifiers Five Fold Result

In Figure 11, the f measure for Naive Bayes is 80%, 79% for Max Entropy and 78% for SVM for five fold.

The performance measures for classifiers varied greatly when doing a five fold cross validation compared with the single fold cross validation. The reason could be that, for single fold cross validation, since we are partitioning the dataset into 75% training dataset and 25% test dataset and this test dataset is very small(only 0.25 of the total dataset) in single fold. So there will be lot of variation in the performance estimate for different samples of data. Whereas, in 5 fold cross validation, this variation is greatly reduced due to the averaging of the results of the 5folds. This makes the performance less sensitive to the partitioning of the data. Also, the partition size of 75% training and 25% testing dataset can also be varied and the results from them can be averaged by k, to get even better results that is insensitive to variation in the dataset.

The dataset, results and code for the project can be viewed at the project repository in https://github.com/tilaprimera/Twitter-Sentiment-Analysis-.

5. Future Work

The classification techniques that we used, have given us considerably good results, but there are certain areas where we can improve our system. we are planning to include those in our future analysis.

1) **Neutral Tweet handling:** The algorithm designed can classify a tweet as positive or negative but neutral tweet is also a very important genre in sentiment analysis. Like for a sentence The weather is very hot. How do we classify this tweet?
2) **Semantics:** How do we translate the perspective of a user? For Example: *Barcelona beats Real Madrid :D* This tweet is positive for Barcelona FC supporters but otherwise for Real Madrid supporters. Semantics can come handy in this case. Using a semantic role labeler may indicate which noun is mainly associated with the verb and the classification would take place accordingly.
3) **Bigger Dataset:** A training dataset with millions of data can give us a better result.
4) **Internationalization:** Sentiment Analysis of the tweets written in different languages other than English will help us cover more language area in the field of sentiment analysis.

6. Conclusion

The unigram model for twitter sentiment classfication is efficient [7] computationally but we can improve on the performance of the classifier using weighted unigrams. The words such as “very”, “extremely” can be assigned more weights. In the similar manner, capslock words can be interpreted as strong sentiment of anger (negative) or excitement (positive) and weights to those words can be applied accordingly. Read[2] suggests assigning weights to emoticons as well, since they are fairly easy to interpret as positive or negative sentiments. Although computationally intensive, bigram model can be used to further improve on the accuracy. Naive Bayes Classifier has outperformed the other classifers Maximum Entropy and SVM. We attribute this result to the classifier's performance towards text classification [8].

Acknowledgments

The authors would like to thank Prof. Dr. Abdul Serwadda for his valuable suggestions.

References

[1] Alec Go, Richa Bhayani, Lei Huang. Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford Digital Library Technologies Project, 2009.

[2] Read, Jonathon. "Using emoticons to reduce dependency in machine learning techniques for sentiment classification."Proceedings of the ACL student research workshop. Association for Computational Linguistics, 2005.

[3] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques."Proceedings of the ACL02 conference on Empirical methods in natural language processingVolume 10. Association for Computational Linguistics, 2002.

[4] Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." LREc. Vol. 10. No. 2010. 2010.

[5] Jiang, Long, et al. "Targetdependent twitter sentiment classification." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies Volume 1. Association for Computational Linguistics, 2011.

[6] http://www.nltk.org/book/ch06.html

[7] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1-16.

[8] anning, Christopher D., and Hinrich Schtze. Foundations of statistical natural language processing. Vol. 999. Cambridge: MIT press, 1999.