Twitter Sentiment Analysis Using Classification and Regression (CART) and Random Forest

LCR Paquit[1], VB Calag[2], CG Manliquez[3], AR Mesa[4]
Department of Mathematics, Physics and Computer Science
College of Science and Mathematics, University of the Philippines Mindanao, Davao City 8000, Philippines
Emails: lrpaquit@up.edu.ph[1], vbcalag@up.edu.ph[2], cgmanliquez@up.edu.ph[3], armesa@up.edu.ph[4]

Abstract

Machine learning algorithms are commonly used in sentiment analysis. The basic task of sentiment analysis is to classify the polarity of the given text whether the sentiment expressed is positive or negative. The focus of the study is to explore the performance of Classification and Regression Tree (CART) and Random Forest (RF) on Twitter sentiments that are labeled using distant supervised learning classifiers, and compare their performance against known Machine Learning algorithm such as Naïve Bayes (NB), Maximum Entropy (MaxEnt), and Support Vector Machine (SVM). The training data consist of random sample sets of 4,000, 8,000 and 12,000 extracted from a pool of 1.6 million tweets with a balanced number of positive and negative tweets. MaxEnt, SVM and RF performances are superior than NB and CART. MaxEnt, SVM and RF performed better with more than 70% accuracy while NB and CART have 64.9% and 51.8%, respectively. The better performing classifiers were used to classify a real tweets extracted using Twitter API. The tweets annotated by human were compared to machine generated classifications which perform similarly except for RF which has an accuracy below 70%. The benefit of this study is to reduce the amount of human supervision to labeled data and use only a small sample data for a fast, low processing cost and limited resources in performing sentiment analysis.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis*.

General Terms

Algorithms, Performance, Experimentation

Keywords

Emoticons, machine learning algorithm, random forest, sentiment analysis, twitter

1. INTRODUCTION

Over the years, the growth of Web 2.0 has been very explosive [33]. Several types of social media like blogs, discussion forums, review websites and community websites can be very useful to determine public opinion or sentiment towards a certain topic. The field of study that analyzes these sentiments or opinions is called sentiment analysis and can be interchangeably referred to as opinion mining [12].

Twitter is a popular micro-blogging service that gain exponential growth over the recent years. It has millions of users including celebrities and media personalities. It allows users to create status messages called "tweets". These tweets are limited to only 140 characters per posting allowing people to share easily quick updates and interest with brevity in their expressions [1].

We compare the performance and accuracy of sentiment analysis of tweets with distant supervision using Classification and Regression Trees (CART) and Random Forest to the results of sentiment analysis using Naïve Bayes, Maximum Entropy, and Support Vector Machine (SVM). We also implement preprocessing on the training and testing data. We train the data by applying CART and Random Forest on each subset of the tweets for classification under different training set sizes. Then, interpret and evaluate the result of prediction of each classifier. And apply the best performing classifiers to a real Twitter dataset.

The remainder of this paper is organized as follows. In section 2, we discuss related work on sentiment analysis and different approaches on sentiment classification. We also discuss application of sentiment analysis on Twitter. In section 3, we describe the methods on collecting data, pre-processing procedures and machine learning approaches. After presenting our implementation, evaluation and experimental result in section 4, we conclude the paper in section 5.

2. RELATED WORK

Sentiment analysis is the computational study of opinions, sentiments and emotions expressed in text. It uses natural language processing, text analysis and computer linguistic to categorize and extract subjective information in the source text. There exist many studies that explore sentiment analysis which deal with different levels of analyzed text, including word or phrase level [36], sentence level [10], document level [14, 28], and at user level [24, 31] sentiments. Word level sentiment analysis studies the orientation of words or phrases in the text and their effect on the overall sentiment while sentence level explores sentences that express a single sentiment and define its orientation as a whole. The document level sentiment analysis is viewing the whole document's general sentiment and user level sentiment looks for the possibility that connected users on the social network may be more likely to hold a similar opinion.

The basic task in sentiment analysis is classifying the polarity of a given text whether the opinion expressed on a single issue is classified as one of two opposing sentiments [13]. Examples of polarity classifications are "negative or positive", "thumbs up or thumbs down" and "like or dislike." Early works in polarity classification include Turney [28] and Pang et al. [14] who applied a different approach in identifying the polarity of product reviews and movie reviews, respectively. Another task of sentiment analysis which has been researched the most is classifying a text

Conference, June 1–2, 2016, Davao City, Davao del Sur, Philippines.

into one of two classes: objective or subjective [15]. This task determines whether a sentence expresses an opinion or not. Distinguishing between subjective and objective help classify the sentiment. Besides, there are texts that might have a polarity without necessarily holding an opinion. One good example is a news headline or a news article. The need for a more fine-grained approach to sentiment analysis has led to aspect-based (or feature-based) sentiment analysis. The task of an aspect-based is to determine the opinions or sentiments expressed on different features or aspects of entities (e.g. laptop, smartphone, digital camera or a restaurant). An aspect is an attribute or part of an entity; for example, the price of a laptop, the battery life of a smart phone, the picture quality of the camera or the service for a restaurant. Different aspects can generate different sentiment responses, for example, a restaurant has nice ambiance but the food is bad. More detailed discussion about this level of undertaking in sentiment analysis is found in Liu's work [12].

2.1 Sentiment Classification Approaches

Classification is a step in sentiment analysis that can be described as a process in which we predict qualitative response or in this case classifying polarity of text. There are many classification techniques or classifiers that can be used to predict qualitative response or class of a text. Sentiment classification techniques can be divided into three existing approaches namely; lexicon-based approach, machine learning approach and linguistic analysis [26].

Lexicon-based approach can be divided into three main approaches – manual, dictionary-based and corpus-based. On the other hand, machine learning approach is classified into two main categories unsupervised approach and supervised approach. One of the simplest unsupervised learning methods is K-means. The supervised learning methods in sentiment analysis can be subdivided into four groups. First is the linear-based classifier and two known classifiers under this category are Support Vector Machine and Neural Network. Second is the probabilistic classifiers and the two known probabilistic classifiers are Naïve Bayes and Maximum Entropy. Third is the decision trees and the most notable decision tree classifiers are Classification and Regression Tree and Random Forest. And the fourth is the rule-based classifiers

2.1.1 Lexicon-Based Approach

A lexicon-based approach predicts sentiment of review text using a predefined list or corpus of words with a certain polarity. Review text is classified by calculating and averaging the polarity score of individual words in sentences. While using a lexicon based approach in sentiment classification, there are factors that need to be considered like word position, word relationship, and negation handling. Among the lexicon based approaches, the three main approaches are the manual approach, dictionary-based approach and corpus-based approach. The manual approach is very time consuming thus it is usually combined with the either of the two automated approaches [12]. Dictionary-based approach finds sentiment seed words and then finds their synonyms and antonyms. One major disadvantage of this approach is its inability to find sentiment words with specific context [3]. Corpus-based approach has a seed list of sentiment words and then finds other sentiment words in the large corpus in the same context. It helps solve the problems of finding sentiment word with context specific orientation. A major disadvantage is that this approach alone is not as effective because it is difficult to prepare huge corpus, unlike dictionary-based approach which covers all the words of English [21].

2.1.2 Linguistic Approach

The linguistic approach uses the syntactic characteristics of the words or phrases, the negation and the structure of the text to determine the text orientation. This approach is usually combined with a lexicon-based method. Thet et al. [27] proposed a linguistic approach for sentiment analysis for movie reviews on discussion boards. In their study, they generate dependency tree and splits the sentence into clauses for each review. Then it determines the contextual sentiment score for each clause using grammatical dependencies of words and the prior sentiment score of the words derived from SentiWordNet and domain specific lexicon. Negation handling was tackled in the study but automatic aspect extraction was not.

2.1.3 Machine Learning Approach

Machine learning approach is not dependent on lexicon dataset, instead it uses a training set and a test set to build the classification model. This approach allows adaptability, especially in the ever changing social network language lexicons. Machine learning approach develops a classification model using a training set, which tries to classify the input feature vectors into corresponding polarity. A test set is used to validate the model by predicting the polarity of unseen feature vectors as outlined in the training set. D'Andrea et al. [6] stated that machine learning approach uses supervised and unsupervised method for sentiment classification.

2.1.3.1 Unsupervised Approach

Unsupervised methods are used when it is difficult to find labeled training documents but it is easy to collect unlabeled documents. It does not need a training set in order to extract an information [32]. The learner uses unlabeled set of examples that includes learning patterns in the input data with no specific output values are provided. Corpus classified using unsupervised learning methods can also be used as the training and testing data in supervised learning method. An earlier work on this approach was studied by Turney [28]. He explains a simple method to do unsupervised sentiment analysis using only the words "excellent" and "poor" as a set seed then uses the mutual information of other words with these two adjectives to achieve an accuracy of 74%.

2.1.3.2 Supervised Approach

Machine learning uses supervised approach when there is a finite set of classes (such as positive and negative). This approach needs two sets of data: the training set and test set. The training set is used by a classifier to learn different properties of document and test set is used to evaluate or validate the performance of the classifier. The existing supervised learning methods in sentiment analysis can be subdivided into four groups of classifiers: linear-based classifiers, probabilistic classifiers, rule-based classifiers and decision tree classifiers

Linear-based classifiers attempt to determine good linear separators between different classes. Two of the most well-known linear classifiers are Support Vector Machine (SVM) and Neural Network (NN). SVM is widely used for text categorization [14]. It seeks a decision surface to separate training data point into two classes and make decision based on support vectors. SVM have the potential to handle large feature spaces with high number of measurements. On the other hand, NN approach is based on a collection of natural or artificial neurons uses for mathematical computational model analysis. Neuron is the basic unit of a neural network and neural networks consist of many such neurons.

Probabilistic classifiers are model mixtures which presume that each class is a mixture of components. Each mixture provides a probability of sampling a particular term for that component. These classifiers are also called generative classifiers and the two predominant classifiers are Naïve Bayes (NB) classifier and Maximum Entropy (MaxEnt) classifier [8]. NB is based on Bayes Theorem, which calculates the probability of a sentiment based on which polarity it belongs (positive, negative or neutral). This classifier basically assumes that all features are independent from each other. MaxEnt on the other hand, is a feature-based probability distribution estimation technique used for various natural language processing purposes, such as sentiment classification. The main principle of maximum entropy classifier is if there is less information on the data, distribution should be as uniform as possible.

In rule-based classifier, given a training set, uses a rule generator to generate a set of rules and a set of patterns to represent the test sample and used the rule set derived from the training set to classify the test sample. This approach considers a collection of documents as a mining field from which a set of patterns can be extracted, optimized and stored in an efficient way for pattern-query matching. The patterns are extracted by using either set of predefined templates or heuristic method.

2.1.3.2.1 Decision Trees

These are one of the most widely used approaches in machine learning algorithms. One of the reasons for their popularity is because they can easily adapt to any type of data. They are supervised machine learning that gives a hierarchal decomposition of training data in which a condition on attribute value is used to divide the data. The condition can be presence or absence of one or more words. Division of data is recursively done until a leaf node contain certain minimum number of records which are used for the purpose of classification [25]. Decision trees have been used in many applications in speech and language processing. Most notable decision tree classifiers are: Classification and Regression Tree and Random Forest.

2.1.3.2.1.1 Classification and Regression Tree

It is used to refer to the two main type of decision trees used in text mining, Classification and Regression. This approach produces either classification tree or a regression one depending on whether the dependent variable is categorical or numeric, respectively. CARTs are formed by an ensemble of rules based on variables in the modeling data set. Rule-based on variables' values are selected to get the best split to differentiate observations based on the dependent variable. Once a rule is selected and branches a node into two, the same process is applied to each child node. Branching out stops when CART detects no further gain can be made or some preset stopping rules are met. Each branch of the tree ends in a terminal node. Each observation falls into one and exactly one terminal node, and each terminal node is uniquely defined by a set of rules [35].

2.1.3.2.1.2 Random Forest

It is a robust classifier introduced by Breiman [17] that can handle categorical input such as sentiments. It is a collection of a learning method for classification that works by constructing a large number of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. It produces multi-altitude decision tree at the inputting phase and generate results in the form of multiple decision trees. The collinearity between trees is reduced by randomly selecting trees and thus, the prediction accuracy increases which in turns make it more efficient. The predictions are made by aggregating the predictions of various collections of data sets. There has been a widespread application and real life examples using random forest but there is no single type of random forest data sets. They can vary from any kind of applications like medical as well as general data sets.

2.2 Distant Supervision Approach

Training a supervised machine learning classifier usually involves an extensive amount of manual process in obtaining labeled training dataset. This labeled training dataset is expensive to produce and thus limited in quantity. In order to reduce the amount of human supervision in sentiment analysis tasks, a learning technique called distant supervision is utilized. The main idea of distant supervision is to make use of a "weakly" labeled training set, where labels are considered to be "noisy" when obtained based on the heuristic task or side information [5]. The idea was formally established in the study of Mintz et al. [7] and coined the term distant supervision. Read [18] were first to consider the existence of specific emoticons as noisy labels to classify sentiment polarity of a document. The same technique is used by Go et al. [30] and Pak and Paroubek [29] who build noisy labeled training set of Twitter posts for sentiment analysis.

2.3 Sentiment Analysis on Twitter

Applying sentiment analysis on Twitter is the upcoming trend with researchers recognizing the challenge and its potential applications. Twitter is a good venue for social network analysis with its unique limitation that a user can only enter 140 characters each tweet [9]. The challenges that are distinct to this problem are largely attributed to the informal tone of the tweets. A tweet can contain a significant amount of information in a very compressed form, and simultaneously carry positive and negative sentiments. Pak and Paroubek [29] cited that the use of twitter as a corpus for sentiment analysis have basis. First, it is a valuable source of people's opinion since it is used by different people to express an opinion on different topics. Second, twitter contains an enormous number of tweets and it grows every day. The collected corpus can be arbitrarily large. Third is audience varies from regular users to celebrities, companies, politicians. And it is represented from many countries worldwide thus it is possible to collect users from different social and interest groups.

Some of the early studies on sentiment analysis of Twitter data are by Go et al. [30], Bermingham and Smeaton [4] and Pak and Paroubek [26]. Go et al. [27] conducted distant supervised learning study on tweets using the emoticons (e.g. "☺", "☹", etc.) as markers of positive and negative tweets. Their study used the same idea as employed by Read [17] to generate corpus using emoticons as labels for positive and negative sentiments. They built models using Naïve Bayes, MaxEnt, and SVM, and they find SVM outperforms other classifiers. Pak and Paroubek [26] also performed similar distant learning model though they perform different classification task which was classifying tweets in terms of subjectivity and objectivity. And they obtain the best result using Naïve Bayes.

3. MATERIALS AND METHODS

3.1 Data Collection

The Stanford Sentiment 140 Tweet Corpus[1] data, collected by Go et al. [30] between the 6th of April and the 25th of June 2009 were used. The training set consisted of 1.6 million general tweets with the same number of positive and negative tweets labeled, and its test set of manually annotated tweets consisted of 177 negative and

[1] http://help.sentiment140.com/for-students

182 positive tweets. These data were stored in a CSV (Comma Separated Values) file format. The tweets were labeled as 0 for negative sentiments and positive sentiments were labeled as 4. Due to the limitation of the computer machine used in this study, it was not feasible to use the entire pool of training set from Sentiment 140 data. Hence, a stratified sample of sizes 4,000, 8,000 and 12,000 tweets were extracted to be used as training sets for this study.

A real Twitter dataset was also extracted using a Twitter Application Programming Interface (API)[2], which consists of 500 tweets. The data was collected on the 31st of March 2016 using the keyword "Duterte". Duterte is a controversial and most-talked about presidential candidate in the Philippines at the time of this study. It was decided to use that keyword because it was the trending topic in Twitter at that time and the researchers believed that it would be easier to get a mix of positive and negative sentiment using the keyword. In order to collect tweets, a Twitter application was needed to setup therefore, it requires to create a Twitter account. The credentials needed to authenticate and access the Twitter API were the consumer key, consumer secret, access token and access secret.

Initially, 500 tweets were extracted using the keyword "Duterte". Retweets were removed from the list. Retweeting is the process of reposting a user's tweet to another account. The tweets were manually annotated as positive or negative. Tweets with no sentiments were labeled as neutral. Therefore, the final list of the real Twitter dataset consisted of 132 tweets with 97 positive and 35 negative tweets, manually annotated by a person.

3.2 Pre-processing Tweets

Pre-processing the tweet is the process of cleaning and preparing the tweet for classification. Tweets usually contain lots of noise and uninformative text that do not contribute to its sentiment. Many tweets include URLS, tags to other users, or symbols that have no meaning. To accurately obtain a tweet's sentiment, the noise that would not help in the training and evaluation process was removed.

Several pre-processing steps were made to clean the tweets and these include:
1. Removing emoticons from tweets. In the study of Go et al. [30] they treat emoticons as noisy labels because it had a negative impact on the accuracies of Machine Learning classifiers. Stripping out emoticons causes the classifier to learn from other features and if the test data contains an emoticon, it does not influence the classifier because it was not part of the training data. These emoticons are listed in Table 1.

Table 1. List of Emoticons as enumerated by Go et al. [30]**.**

Emoticons mapped to :)	Emoticons mapped to :(
:) :-) :) :D =)	:(:-(: (

2. Removing <@username>. Usernames was removed because this did not add sentiment in a tweet.
3. Removing URLs. It is common to find URLs in tweets. All strings that describe links or hyperlinks were removed.
4. Remove numbers. Numbers in general, do not represent a sentiment thus, numbers were removed in tweets.
5. Tokenization. Tweets were split up into terms or tokens by spaces forming a list of individual words per tweet.
6. Case normalization. The entire data were changed to lower case. The reason for this was because the classifier will treat words like, amazing, Amazing, amAzinG and AMAZING as different words. This method greatly reduced the sparsity of our data set.
7. Removing punctuations. All punctuations and other symbols were removed as they add no value to the text.
8. Strip multiple whitespaces. Strip extra whitespace from a tweet and characters were collapsed to a single space.
9. Removing stop words. Stop words are words that do not add meaningful content to the tweet. Removing them reduced the space of the items significantly in the training and testing set. Example of such words include, the, a, and, to, of, etc.
10. Remove sparse terms. Terms that appear very often were removed. This pre-processing step was necessary to reduce computational cost because the more the terms means it takes longer to build the classifier.
11. Stemming. It is the process of removing prefix and suffix leaving the stem or the root of the word. For example, the set of words read, reader, readers, and reading all will be reduced to the root word read. SnowBallC stemmer which is a package in R was used to implement this process.

3.3 Machine Learning Methods

We tested different machine learning classifiers: Naïve Bayes, Maximum Entropy, Support Vector Machine, CART and Random Forest.

3.3.1 Baseline

As a baseline, the three traditional machine learning techniques conducted by Go et al. [30] was the basis of this study. The three classifiers were: Naïve Bayes, Maximum Entropy, and Support Vector Machine.

Naïve Bayes classifier is a probabilistic classifier based probability models that incorporate strong independence assumptions among the features. Assign to a given tweet d the class c* =〚argmax〛_c P_NB (c|d).

$$P_{NB}(c|d) = \frac{(P(c)\sum_{i=1}^{m} P(f|c)^{n_i(d)})}{P(d)} \qquad [1]$$

In equation [1], f represents a feature and n_i(d) represents the count of feature f_i found in tweet d. There are total of m features. Parameters P(c) and P(f|c) are obtained through maximum likelihood estimates.

Maximum Entropy's main principle is to find the best probability distribution among prior test data. It yields maximal entropy information which gives proper distribution. Maximum entropy classifiers are generally used as alternatives to naive Bayes classifiers because they do not assume statistical independence of the random variables (commonly known as features) that serve as predictors. However, learning in such a model is slower than for a Naïve Bayes classifier, and thus may not be appropriate given a very large number of classes to learn.

$$P_{ME}(c|d,\lambda) = \frac{\exp\,[\sum_i \lambda_i f_i(c,d)]}{\sum_{c'} \exp\,[\sum_i \lambda_i f_i(c,d)]} \qquad [2]$$

In equation [2], c is the class, d is the tweet, and λ is a weight vector. The weight vector decides the significance of a feature in classification. A higher weight means that the feature is a strong indicator for the class. The weight vector is found by numerical optimization of the lambdas so as to maximize the conditional probability.

[2] https://dev.twitter.com/overview/api

Support Vector Machine maximizes the distance between the hyperplane and “difficult points” close to decision boundary. If there are no points near the decision surface, then there are no very uncertain classification decisions. Maximizing the margin is a problem of constrained optimization, which can be solved by Lagrange multiplier〖 α〗_i:

$$w = \sum_{i=1}^{\#sv} \alpha_i y_i x_i^{sv}, \alpha_i \geq 0 \qquad [3]$$

In equation [3], α_i =0, x_i^ has no influence on the hyperplane. α_i> 0, x_i^ determines the separate hyperplane. These values are called support vectors and they are the only vectors contributing to w. Once the SVM is built, classification of test tweets simply involves determining which side of w’s hyperplane they fall on.

3.3.2 Classification and Regression Tree (CART)

CART analysis is a form of binary recursive partitioning. The term “binary” implies that each group of patients, represented by a “node” in a decision tree, can only be split into two groups. Thus, each node can be split into two child nodes, in which case the original node is called a parent node. The term “recursive” refers to the fact that the binary partitioning process can be applied over and over again. CART creates a flowchart based classifier as illustrated in Figure 1.

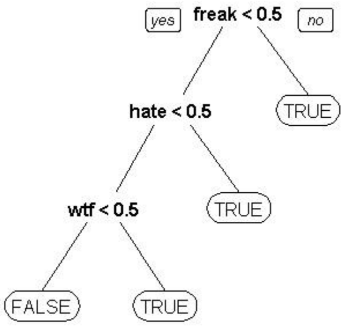

Figure 1. Simple CART structure.

In pseudocode, the algorithm for building CART as presented by Kotsiantis et al. [11] is shown as follows:

1. Check for base cases
2. For each attribute a
    a. Find the feature that best divides the training data
3. Let a_best be the attribute with the best feature in training data
4. Create a decision node that splits on a_best
5. Recur on the sub-lists obtained by splitting on a_best, and add those nodes as children of node.

We used RtextTools package in R to train the tweet corpus into CART classifier. RTextTools extended the package tree that followed Breiman et al. (1984) algorithm, it provided re-implementation of the tree [19].

3.3.3 Random Forest

Bootstrapping helps to reduce the noise in data, and the random selection of features further reduces the risk of over-fit, since each tree only focus on a subset of features. Random forests are built using an ensemble of decision tree classifiers, each tree is trained using a random sampled subset of the training data. The algorithm will only consider splitting from a random subset of the chosen features. The major disadvantage with using a single tree is that it has high variance which means that the arrangement of the training data and features may affect its performance. Getting the average over an ensemble of trees can reduce the variance of the overall classification model. The more decision trees that are averaged the lower the variance will be. And as a result of lowering the variance, it will largely increase the overall performance of the model [17].

Figure 2 depicts a random forest structure. It consist of B decision trees. Each decision tree is trained by a completely random approach. For each decision tree, $tree_B$, the samples are selected randomly from the training sample set. It is a subset of all the training samples. After B trees are trained, the final decision combines all the outputs of $tree_1$, $tree_2$ … $tree_B$ by majority votes among the k decisions.

Figure 3 showed a more detailed pseudocode of Random Forest [2].

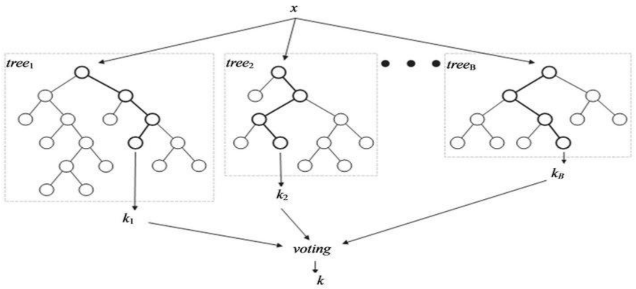

Figure 2. Random forest structure (Lifted from Moore, 2015).

```
To generate c classifiers:
for i = 1 to c do
        Randomly sample the training data D with replacement to produce Di
        Create a root node, Ni containing Di
        Call BuildTree( Ni )
end for
BuildTree(N):
if N contains instances of only one class then
        return
else
        Randomly select x of the possible splitting features in N
Select the feature F with the highest information gain to split on
Create f child nodes of N, N1 ,..., Nf, where F has f possible values ( F1 , …, Ff )        for i = 1 to f do
            Set the contents of Ni to Di , where Di is all instances in N that match Fi
   Call BuildTree( Ni )
 end for
end if
```

Figure 3. Pseudocode for random forest algorithm.

3.3.4 Performance Evaluation

The prediction result and the actual value was compared using a confusion matrix as shown in Figure 4.

		predicted outcome	
		P’	N
actual value	P	True Positive	False Negative
	N	False Positive	True Negative

Figure 4. Confusion matrix.

The performance metrics that was used to evaluate the classification results were precision, recall, F-measure and overall accuracy. These metrics were computed based on the values of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) assigned classes. Precision is the number of true positive out of all positively assigned documents, and it is given by

$$Precision = \frac{TP}{TP+FP} \qquad [4]$$

Recall is the number of true positive out of the actual positive documents, and it is given by

$$Recall = \frac{TP}{TP+FN} \qquad [5]$$

Finally F-measure is a weighted method of precision and recall, and it is computed as

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \qquad [6]$$

Where its value ranges from 0 to 1 and indicates better result the closer it is to 1.

And each classifiers,

$$Overall\ Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad [7]$$

Additionally, a 10-fold cross validation was performed to find the average accuracy and get the best estimate of error. N-fold cross validation means splitting the training set into N sets. One of those set was left out as a test set to measure accuracy, and the N-1 other sets were used to train the classier. This was repeated N total times, one for each separate data partition. Afterwards, the accuracies were averaged and reported.

4. RESULTS AND DISCUSSION

Using a stratified random sampling, 4,000, 8,000 and 12,000 set of tweets were extracted from the pool of training set for building the classifier models. Small and randomized subsets were used in order to ensure that it is within the memory and processing capacity of the computer machine used in conducting the study.

4.1 Pre-processing Implementation

The aim of data pre-processing is to remove any uninformative content from the training data and the input tweets, and to prepare the tweets for classification. The term uninformative content are information within the tweet that will not or less likely to be useful for the machine learning algorithm to assign a sentiment to that tweet. Pre-processing steps did not only simplify the classification task for the machine learning model but it also greatly decreased the processing cost in the training phase. Uninformative contents in the tweets such as username, url, weblinks, punctuations, numbers, stop words were removed from the training set. Uppercase characters were all set to lowercase and multiple whitespaces were collapsed to a single space.

Using pre-processing steps, these allowed more specific word features to be passed into the classification models and hugely reduce processing cost during the training stages. Examples of tweets before and after processing are contained in Table 2. All these were loaded in create_matrix function to create a document-term matrix that was then passed to the classifier implementation stage.

Table 2. Sample of pre-processed tweets.

Raw Data	Processed Data
Graduation is in about 2 weeks...I can't believe it. Everyone's invited!	graduat weeksi cant believ everyon invit
@solarrhyll OH MY GOD, that would be genius, I actually need a couple shot for my port folio	oh god genius actual need coupl shot port folio
@FlissTee Oh, fabulous ta. I'm with the coffee too, obviously, although my cup is not large enough to embrace which is its deficiency.	oh fabul ta im coffe obvious although cup larg enough embrac defici
Bed at 3.30am and then woken at 7.30am when new toy (HP DV2-1035EA) delivered 2 weeks earlier than anticipated http://snurl.com/gcitj	bed woken new toy hp dvea deliv week earlier anticip

4.2 Classifier Performance Evaluation

Each model was evaluated under four criteria; overall accuracy, precision, recall and F-measure. The accuracy is simply the proportion of correctly classified tweets. Precision, recall and F-measure evaluate the performance of classifier on each polarity class. Precision measures the number of instances correctly classified in that class out of all the instances the classifier predicts as that class. Recall on the other hand, measures the number of instances correctly classified in that class out of all the actual instances of that class. The F-score or F-measure indicates the models overall performance of both precision and recall in each polarity, where the highest level of performance is equal to 1 and the lowest is 0 [22]. This evaluation process is focused on selecting the suitable classification model for the dataset. True Positive (TP) is the number of tweets correctly classified as positive while False Positive (FP) are those tweets classified as positive but are actually negative. On the other hand, True Negative (TN) are tweets correctly predicted by the classifiers as negative while False Negative (FN) are tweets wrongly predicted as negative tweets.

Figure 5 shows the performance of the models using different training set sizes. The performance of the models with 4,000 training set size shows that in terms of overall accuracy MaxEnt performed best with 75.2% accuracy followed by SVM with 74.1%. Random forest was evaluated using different numbers of random trees to see whether it affects the models performance. RF* uses 10 random trees, RF** with 20 random trees and RF*** have 200 random trees. Interestingly enough it did increase but only a margin of positive effect. Garnering an accuracy of 70.8%, 72.1% and 72.4% from 10, 20 and 200 random trees, respectively. Compared to other algorithms, NB and CART (55.1% and 51.8%, respectively), did not perform very well. There were substantial improvement on the accuracy of the models using 8,000 tweets as training data except for CART which still have an accuracy of 51.8%. Precision, recall and f-measure, which was calculated for each polarity, improved on each model using the training data. There is a noticeable decline in the performance of the classification models using 12,000 tweets as training data. It could be that by increasing the training set size the classifiers are learning new features but also adding more noise terms that are useless information for building the classification models [18]. Not only did it performs poorly compared to the performance of models with 8,000 training set size but increasing the size also increased the computational overhead to train the model.

From the evaluation, it was discovered that the performance of Naïve Bayes and CART were significantly lower compared to MaxEnt, SVM and Random Forest. Although Naïve Bayes are less likely to over fit the training data with smaller sample size, it needs enough data to understand the probabilistic relationship of each feature in isolation with the output variable. On the other hand, CART tends to over fit the classifier with smaller sample size of training data, while it performs well with larger data. Thus, it did not have the same level of predictive accuracy as compared to other classification method. The same trend was produced by Soni and Mathai [23] on their work.

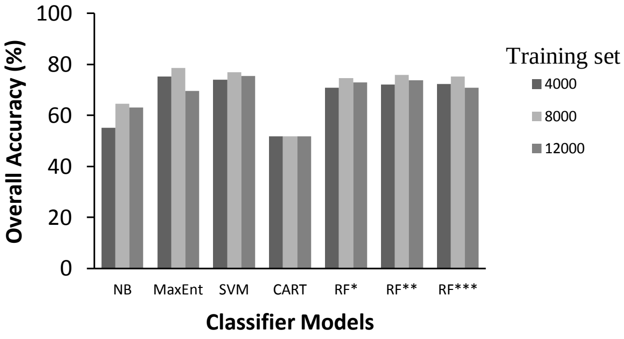

Figure 5. Classifiers accuracy with 4,000, 8,000 and 12,000 training set sizes.

Figure 6 shows the training time to build each model in three different training set sizes. NB, MaxEnt, and CART were trained significantly faster, not more that 6 seconds, as compared to SVM and RF. Generally, increasing the number of random trees used in random forest algorithm also increased the overall accuracy of the algorithm. Although the more trees you add the higher the computational overhead will be, which was not significant for this study.

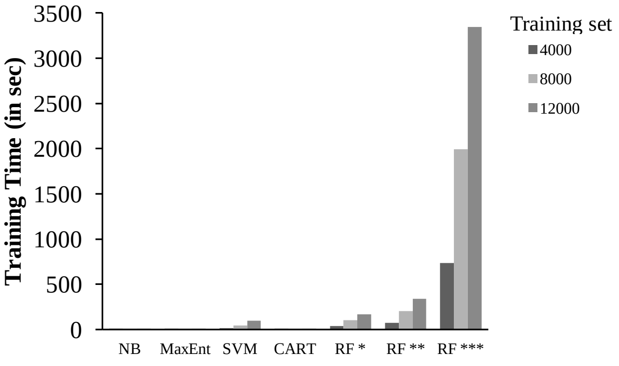

Figure 6. Training time of each model using the 4,000, 8,000 and 12,000 training set.

The three classifiers: MaxEnt, SVM and Random Forest, perform well with over 70% accuracy when predicting the test set. This result was encouraging considering the noise that was likely to be present in the data and the sample size used to train the models were very small compared to the original data which was 1.6 million tweets. There seems to be an apparent inconsistency for the training sizes of 4,000, 8,000 and 12,000 tweets, where the accuracy increases for training set size, then subsequently decreases on the third set. This implies that the classifier was learning features that were useful in classifying the test set, but the addition of more noise and irrelevant texts makes the information redundant. These needed to be investigated and solved in future studies. Some examples of noise taken from the training set are presented in Table 3.

Table 3. Sample noise taken from the training set.

Noise	Example
mixed emotions	"Three days of leisure ends today. Three days of training starts tomorrow. But in the end IT'S ALL FUN ",
spelling mistakes	"@xxlucyappxx u must fulfil ur destiny - lol - mayb next time",
sarcasms	"@carriebubz you have no followers bless xx"

Among the three training sizes, it is apparent that models trained with 8,000 data performed best. MaxEnt, SVM and RF had an accuracy of 78.6%, 76.9% and 75.8%, respectively. Thus, it is reasonable to carry out an in depth evaluation of them. To validate the models effectiveness, an evaluation technique was used called 10-fold cross validation on the training set. This is to validate if the experimental results will be the same as if they were obtained on completely independent test sets. Hence, lowering bias on the training set. The study used 10-fold cross validation instead of 3-fold or 20-fold, because extensive tests on many different datasets have shown that 10-fold cross validation is needed to get the best estimate of error [20]. To do this, first break the data into ten sets of size n/10, then train on nine training set and test on one, repeat 10 time and take mean accuracy. The results on cross validation were tabulated in Table 4 and the resulting mean accuracy were 79.4%, 72.5% and 71.5% on MaxEnt, SVM and RF, respectively.

Table 4. Cross validation values using 10-fold.

N-Fold	MaxEnt (%)	SVM (%)	RF (%)
Fold 1	79.9	72.5	68.7
Fold 2	80.4	72.9	71.9
Fold 3	76.9	74.5	71.1
Fold 4	80.4	73.0	72.8
Fold 5	78.6	71.7	73.3
Fold 6	82.6	73.3	70.6
Fold 7	78.0	70.7	71.5
Fold 8	79.8	72.4	70.0
Fold 9	78.6	73.1	72.3
Fold 10	78.6	71.5	72.4
Mean Accuracy	79.4	72.5	71.5

4.3 Testing the Models in Real Twitter Dataset

In order to be thorough on the three selected best performing models, it was tested out on a real Twitter dataset with 132 tweets. The dataset was first classified manually by a human which resulted in 97 positive tweets and 35 negative tweets. Then the dataset was classified using the three models. The dataset also went through the same pre-processing steps. The manual classification and the generated classification using the models were compared to see how close the results of the models as compared to a human classification. Using the confusion matrix the overall accuracy, precision, recall and F-measure were calculated as shown in Table 5.

Table 5. Real Twitter data classification performance.

Model	TP	FP	FN	TN	Overall Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Polarity
MaxEnt	79	21	18	14	70.5	79.0	81.4	80.2	Positive
						43.8	40.0	41.8	Negative
SVM	87	25	10	10	73.5	77.8	89.7	83.3	Positive
						50.0	28.6	36.4	Negative
RF	69	19	28	16	64.4	78.4	71.1	74.6	Positive
						36.3	45.7	40.5	Negative

When examining the results from the manual classification it can be observed that the accuracy of the models is similar to the results during the training phase. SVM got the closest result to the human classified dataset with 73.5% accuracy. MaxEnt got 70.5% and RF got 64.4%. Based on the F-measure, it shows that the three models seems to perform poorly in classifying negative tweets having to only predict less than half of the manually labeled negative tweets. Note that the data used to train the model was extracted in 2009 with no specific topic or domain. If the training data set used was recent and limited to a particular domain in this case politics, the classifier models may perform better [5].

5. CONCLUSION

During evaluation Naïve Bayes and CART did not perform very well as compared to MaxEnt, SVM and Random Forest. At best, Naïve Bayes got 64.9% and CART produced 51.8% accuracy. On the other hand, MaxEnt, SVM and Random Forest performs well with 78.6%, 76.9% and 75.8%, respectively. A 10-fold cross validation was performed on the training data to test the generalizability of the algorithm. The cross validation reported that the best performing algorithm for the training data used was MaxEnt with 79.4% mean accuracy while SVM got 72.5% and RF got 71.5%.

The three best performing models were used to classify a real twitter dataset with 97 positive and 35 negative tweets. The dataset was first classified by human and then compared to the trained classifiers. The results showed that SVM got the closest to human classified tweets with 73.5%, next was MaxEnt with 70.5% and lastly RF which was significantly lower than the two got 64.6%.

Although CART did not perform well compared to other learning algorithms, Random Forest seems to compete very well with other algorithms. In terms of accuracy, RF is far more superior than NB but relatively lower to MaxEnt and SVM. Overall, the study still present an interesting result using only a sample training data to train the models. The benefit of this study is to reduce too much dependence on labeled data which is very expensive to acquire and also using only a small sample data for a faster, less computational overhead and resource-limited processing of the data.

6. REFERENCES

[1] 7 Things You Should Know About Twitter: 2007. *https://net.educause.edu/ir/library/pdf/ELI7027.pdf*. Accessed: 2013-11-15.

[2] Anderson, G. 2008. *Random relational rules*. University of Waikato.

[3] Angulakshmi, G. and ManickaChezian, D. 2014. An Analysis on Opinion Mining: Techniques and Tools. *International Journal of Advanced Research in Computer and Communication Engineering*. 3, 7 (2014), 7483–7487.

[4] Bermingham, A. and Smeaton, A.F. 2011. On using Twitter to monitor political sentiment and predict election results. *Psychology*. (2011), 2–10.

[5] Broß, J. 2013. *Aspect-Oriented Sentiment Analysis of Customer Reviews Using Distant Supervision Techniques*. Freie Universität Berlin.

[6] D'Andrea, A. et al. 2015. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*. 125, 3 (2015), 26–33.

[7] Distant supervision for relation extraction without labeled data: 2009. *http://jan.stanford.edu/pubs/mintz09.pdf*. Accessed: 2016-05-03.

[8] Dudhat Ankitkumar, M. et al. 2015. A survey on machine learning techniques in sentiment analysis. *International Journal of Futuristic Machine Intelligence & Application*. 1, 2 (2015), 1–5.

[9] Hemalatha, I. et al. 2013. Sentiment Analysis Tool using Machine Learning Algorithms. *International Journal of Emerging Trends & Technology in Computer Science*. 2, 2 (2013), 105–109.

[10] Jagtap, V.S. and Pawar, K. 2013. Analysis of different approaches to sentence-level sentiment classification. *International Journal of Scientific Engineering and Technology*. 2, 3 (2013), 164–170.

[11] Kotsiantis, S. et al. 2007. Supervised machine learning: A review of classification techniques. *Informatica*. 31, (2007), 249–268.

[12] Liu, B. 2012. *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers.

[13] New avenues in opinion mining and sentiment analysis: 2013. *http://sentic.net/new-avenues-in-opinion-mining-and-sentiment-analysis.pdf*. Accessed: 2016-11-15.

[14] Pang, B. et al. 2002. Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Philadelphia, 2002), 79–86.

[15] Pang, B. and Lee, L. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval*. 2, 1-2 (2008), 1–135.

[16] Prabowo, R. and Thelwall, M. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics*. 3, 2 (2009), 143–157.

[17] Random forest: 2001. *https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf*. Accessed: 2016-11-15.

[18] Read, J. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. *Proceedings of the ACL Student Research Workshop* (2005), 43–48.

[19] Recursive partitioning and regression trees: 2015. *https://cran.r-project.org/web/packages/rpart/rpart.pdf*. Accessed: 2015-12-01.

[20] Refaeilzadeh, P. et al. 2009. Cross-validation. *Encyclopedia of Database Systems*. Springer US. 532–538.

[21] Sentiment analysis and subjectivity: 2010. *http://people.sabanciuniv.edu/berrin/proj102/1-BLiu-Sentiment Analysis and Subjectivity-NLPHandbook-2010.pdf*. Accessed: 2015-11-22.

[22] Sokolova, M. et al. 2006. Beyond accuracy, F-Score and ROC: A family of discriminant measures for performance evaluation. *Advances in Artificial Intelligence*. 4304, c (2006), 1015–1021.

[23] Soni, R. and Mathai, K.J. 2015. Improved Twitter sentiment prediction through " cluster-then-predict model ." *International Journal of Computer Science and Network*. 4, 4 (2015), 559–563.

[24] Taking sides: User classification for informal online political discourse: 2008. *http://bulba.sdsu.edu/~malouf/papers/Takingsides.pdf*. Accessed: 2015-11-20.

[25] The text mining handbook: advanced approaches in analyzing unstructured data: 2007. *https://wtlab.um.ac.ir/images/e-library/text_mining/The Text Mining HandBook.pdf*. Accessed: 2015-11-23.

[26] Thelwall, M. et al. 2011. Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology*. 62, 2 (2011), 406–418.

[27] Thet, T.T. et al. 2009. Sentiment analysis of movie reviews on discussion boards using a linguistic approach. *Proceedings of the 1st international CIKM* (2009), 81–84.

[28] Turney, P. 2002. Thumbs Up or Thumbs Down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (Philadelphia, 2002), 417–424.

[29] Twitter as a corpus for sentiment analysis and opinion mining: 2010. *http://lrec-conf.org/proceedings/lrec2010/pdf/385_Paper.pdf*. Accessed: 2015-11-24.

[30] Twitter Sentiment Classification using Distant Supervision: 2009. *https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf*. Accessed: 2015-11-13.

[31] User-level sentiment analysis incorporating social networks: 2011. *http://www.cs.cornell.edu/~chenhao/pub/user-level-sentiment-analysis-incorporating-social-networks.pdf*. Accessed: 2015-11-20.

[32] Veeramani, S. and Karuppusamy, S. 2014. A survey on sentiment snalysis technique in web opinion mining. *International Journal of Science and Research*. 3, 8 (2014), 1776–1780.

[33] Web 2.0 - The evolution and growth of the interactive internet: 2008. *http://www.hcltech.com/hcl-research/enterprise-transformation-services/web-20-evolution-and-growth-interactive-internet*. Accessed: 2015-11-13.

[34] What counting jelly beans can teach us about machine learning: 2015. *http://www.galvanize.com/blog/machine-learning-and-the-wisdom-of-crowds/*. Accessed: 2015-12-04.

[35] Wilkinson, L. 2004. Classification and regression trees. *Systat*. 35–56.

[36] Wilson, T. et al. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (Vancouver, 2005), 347–354.