THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

SENTIMENT ANALYSIS
OF
TURKISH TWEETS

Master Thesis

ERKUT EVİRGEN

ISTANBUL, 2016

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

SENTIMENT ANALYSIS
OF
TURKISH TWEETS

Master Thesis

ERKUT EVİRGEN

Supervisor: PROF. DR. ADEM KARAHOCA

ISTANBUL, 2016

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

Name of the thesis: Sentiment Analysis Of Turkish Tweets
Name/Last Name of the Student: Erkut Evirgen
Date of the Defense of Thesis: March 25, 2016

The thesis has been approved by the Graduate School of Natural And Applied Sciences.

Assoc.Prof.Dr. Nafiz ARICA
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Arts.

Asst.Prof.Dr. Tarkan Aydın
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Arts.

Examining Comittee Members Signature

Thesis Supervisor
Prof.Dr. Adem KARAHOCA -----------------------------------

Member
Prof.Dr. İbrahim PINAR -----------------------------------

Member
Asst.Prof.Dr. Dilek KARAHOCA -----------------------------------

ACKNOWLEDGMENTS

I am deeply grateful to Mr. Prof.Dr.Adem Karahoca who has shaped my vision so much. Having worked with him has given me chance to see closely his analitical skills and experience his common sense. I do feel very lucky.

I would also like to extend my thanks to my friend Yusuf Tok who never stopped illuminating my view all along with his mind and knowledge.

My final and very essential thanks are for my loving wife Gokce Ince Evirgen without whose endless support, patience and wisdom I could not accomplish.

ISTANBUL, 2016 ERKUT EVIRGEN

ABSTRACT

SENTIMENT ANALYSIS OF TURKISH TWEETS

Erkut Evirgen

Computer Engineering

Thesis Supervisor: Prof.Dr. Adem KARAHOCA

March 2016, 37 Pages

This thesis proposes a general frame in R programming language; to act as a gateway for the analysis of the tweets that portray emotions in a short and concentrated format. The target tweets include brief emotion descriptions and words that are not used with a proper format or grammatical structure.

Majority of the work constituted in Turkish includes the data scope and the aim of preparing a data-set. There is no concrete and usable work done on Turkish Tweet sentiment analysis as a software client/web application. This thesis is a starting point on building up the next steps. The aim is to compare five different common machine learning methods: Support Vector Machines, Random Forests, Boosting, Maximum Entropy and Artificial Neural Networks.

Keywords: Support Vector Machines (SVM), Random Forests, Boosting, Maximum Entropy, Artificial Neural Networks

ÖZET

TÜRKÇE TWEETLERİN DUYGU ANALİZİ

Erkut Evirgen

Bilgisayar Mühendisliği

Tez Yöneticisi: Prof.Dr. Adem KARAHOCA

Mart 2016, 37 Pages

Bu tezde, kısa ve öz şekilde duyguların belirtildiği ve kelimelerin doğrudan işlenebilecek kadar düzgün formatlı olmadığı Türkçe tweetlerin R programlama diliyle işlenebilmesi ve bu konuda bir başlangıç noktası olması açısından genel bir çerçeve önerisinde bulunulmuştur.

Bunun yanı sıra özellikle Türkçe alanda yapılan çalışmaların çoğu, analiz edilecek verinin kapsamı ve data set oluşturmak üzerine hazırlanmıştır. Türkçe sentiment analizi üzerine sağlam ve kullanılabilir bir web veya istemci uygulaması henüz yoktur. Bu tez bir adım ileri safhada çalışmalar yapılabilmesi için bir başlangıç noktasıdır. Amacı da en yaygın makine (yapay) öğrenme metodları olan Destekçi Vektör Makinası, Rasgele Orman Karar Ağaçları, Boosting, Maksimum Entropi, Yapay Sinir Ağları karşılaştırmalarını yapmaktır.

Anahtar Kelimeler: Destekçi Vektör Makinası, Rasgele Orman Karar Ağaçları, Boosting, Maksimum Entropi, Yapay Sinir Ağları

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	:	Artificial Neural Networks
API	:	Application Programming Interface
AUC	:	Area Under The Curve
BOW	:	Bag Of Words
CART	:	Classification And Regression Tree
K-NN	:	K-Nearest Neighbors
LIBSVM	:	A Library For Support Vector Machines
MAXENT	:	Maximum Entropy
NLP	:	Natural Language Processing
RF	:	Random Forest
ROC	:	Receiver Operating Characteristic
SVM	:	Support Vector Machines

1. INTRODUCTION

Others' opinions have always mattered to mankind. Whether it was to wage wars, or make a simple choice as picking a cola from the local grocery store, we have always looked at what others think about the choice we are about to make. Perhaps it emanates from an inherent conforming-with-the-majority attitude, but the bottom line is, that opinions do matter. More so in today's digital world, where thanks to the reach and penetration of the internet, opinions at a global scale are available. 140 characters.

That is all that it takes today to make a difference. The micro blogging site called Twitter, has fast emerged as one of the most powerful social media sites which can sway opinions.

Sentiment or opinion analysis, has of late emerged one of the most researched and talked about subject in Natural Language Processing (NLP), thanks mainly to sites like Twitter. In the past, sentiment analysis models using Twitter data have been built to predict sales performance, rank products and merchants, public opinion polls, predict election results, political standpoints, predict box-office revenues for movies and even predict the stock market. There are a plethora of start-up companies which have emerged who are very vociferously engaging in sentiment/opinion analysis for maximizing their revenues.

However, gathering opinion or analyzing sentiment in not as straight forward as it seems. Like mentioned, above, it is one of the most challenging problems in NLP.

With the recent surge in the availability of data, companies the world over, are leveraging the power of gaining insights from data to solve real world problems or for achieving business goals. The volume, velocity and variety of data being generated has reached unprecedented rates. Not only has this called for newer platforms like Hadoop, to handle big data, but also new machine learning techniques and algorithms to derive insights from the data. The focus of this project is on one such technique to handle both structured and unstructured data, viz. Sentiment Analysis. Wikipedia defines Sentiment Analysis as "the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials." The basic idea behind Sentiment Analysis is to extract an opinion. And opinions do matter. In today's highly connected world, social networking sites rule the roost. The number of "likes", "dislikes", "retweets", ratings etc. sway the core thinking of human beings. Today, product managers are more concerned about opinions on social networking sites of their products, rather than feedback provided on their site. Movie reviews and restaurant reviews on such social networking sites dictate the amount of profit the movie producer or restaurant owner is likely to rake in.

Sentiment Analysis essentially looks at classifying the polarity of text, emoticons and now days, even images and videos. The aim is to find out whether the source material is positive, negative or neutral. A lot of the sentiments expressed through social networking sites may not get captured by more traditional survey questions. Sentiment Analysis bridges this gap.

What is aimed to explain shortly in this thesis are as follows:

Firstly, it is to compare SVM, Random Forests, Boosting, Maximum Entropy, Artificial Neural Networks, which happen to be some of the most widespread classification algorithms, with one another amongst themselves by using packages of R programming language.

As for comprison criteria; accuracy and precison recall values of syntesized datas will be used. In addition to this all, a compraision will be made for all datas by using ROC Curve. Also, it is aimed that it offers more source and extra value for those who (individiual or institutional) would want to make Sentiment Analysis in Turkish language as they seek answers for such question as " Which machine learning algorithms?"

Along with this, there has been researchs made by using different machine learning algorithms, so here he differences between their results and this thesis' shall be found.

2. LITERATURE AND BACKGROUND

2.1. LITERATURE REVIEW

Carried out a literature review by examining various papers published over the last two years related to Sentiment Analysis published in various IEEE publications and conferences.
A comparitive analysis on techniques used for sentiment analysis (Ghag and Shah 2013) “shows techniques utilizing both lexicon and non-lexicon based approaches for polarity identification”. The paper also refelects on a multilingual approach and concludes by stating that no existing technique is language independent, thereby prompting a case for a generalized Sentiment Analyser.

Another paper (Rui Xia and Li 2015) talks about dual sentiment analysis. The paper proposes a model to handle the polarity shift problem and brings out the inadequacies of the bag of words (BOW) approach. The proposed model creates reversed reviews that are sentiment-opposite to the original reviews, and make use of the original and reversed reviews in pairs to train a sentiment classifier and make predictions.

Similar to the organization of Gmail's inbox, Sahnkar Setty et al (Shankar Setty 2014), propose classification of Facebook news feeds based on sentiment analysis. The proposed model automatically identifies "important" feeds which reduces manual survey work which is done for drawing conclusions on opinions posted on Facebook.
Riyanul Islam (Islam, 2014) describes a procedure of obtaining a unified user rating system for Google Play Store apps by sentiment analysis on written reviews as well as the starred ratings.

The use of sentiment analysis of social media content for forecasting election results (Pakistan general elections 2013) has been shown (Razzaq, 2014). The results obtained have shown remarkable accuracy to the actual outcome of the elections.

The ability to identify opinions in the presence of diverse modalities is becoming increasingly important. This paper (Rosas, 2013) experiments with several linguistic, audio, and visual features, and shows that the joint use of these three modalities significantly improves the classification accuracy, as compared to using one modality at a time.

Another interesting study is done by having Youtube videos with movie reviews in focus. In his study, Wöllmer (Wöllmer 2013) tries to analyse the sentiment in online Youtube videos automatically. He proposes to add audio features that are commonly used in speech-based emotion recognition to the already done analysis based on text. Encoding the emotional force information conducted by the speaker, also is an important aspect in determining the sentiment of the overall video.

The attention on Twitter for sentiment analysis is immense. This paper (see Aliza Sarlan 2014) reports on the design of a sentiment analysis, extracting a vast amount of tweets. Prototyping is used in this development. Results classify customers' perspective via tweets into positive and negative, which is represented in a pie chart and html page. (Bhuta 2014) reviews a number of techniques, both lexicon-based approaches as well as learning based methods that can be used for sentiment analysis of Twitter text.

Use of an SVM classifier combined with a cluster ensemble (Coletta 2014) offers better classification accuracies than a stand-alone SVM. The paper proposes an algorithm which can refine tweet classifications from additional information provided by clusterers, assuming that similar instances from the same clusters are more likely to share the same class label.

VK Singh et al. (Singh 2013) present an experimental work on a new kind of domain specific feature-based heuristic for aspect-level sentiment analysis of movie reviews. The methodology adopted analyses the textual reviews of a movie and assigns it a sentiment label on each aspect. The scores on each aspect from multiple reviews are then aggregated and a net sentiment profile of the movie is generated on all parameters.

Detection of anomalies (Wang 2014) in tweets in a timely manner can be very benificial. In this study, the authors survey existing anomaly analysis as well as sentiment analysis methods and analyze their limitations and challenges. To tackle the challenges, an enhanced sentiment classification method is proposed and discussed.

Beltagy et al. (El-Beltagy 2013) works on the main setbacks an open points that Arabic social media sentiment analysis come across. The case study in the paper the to explores the idea of determining the semantic orientation of Arabic Egyptian tweets. One of the outcomes of the presented study is “an Egyptian dialect sentiment lexicon.”

Duwairi’s study (Duwairi 2015), examines sentiment analysis on tweets in Arabic by using dialectical words.

The paper uses machine learning techniques to determine the polarity of tweets written in Arabic on dialects. Use of dialectical Arabic, which can be found vastly in social media, creates challenges for topical classifications and sentiment analysis.

For the Turkish language, Sinem Demirci (Demirci 2014) has carried out an emotion analysis on Turkish Tweets. Rather than doing a sentiment analysis, the author talks about emotion analysis. She classifies emotions as joy, sadness, anger, fear, disgust and surprise.

Feature selection is implemented through information gain with term count and term frequency-inverse document frequency as the weighing factors. Naive Bayes, Compliment Naive Bayes and k-nearest neighbors have been used. A data set containing more diverse and a greater number of tweets may better generalize the tweets in Turkish. Also the work does not cater for neutral tweets. Elimination of such non-emotional tweets will help in better classification.

Erogul (Erogul, 2009) researched feasibility of the bag of words and creating Turkish polarity scale data. On the other hand, he focused differantiate factors of Turkish and

English data. He did sentiment analysis by using SVM and some common NLP techniques by using Turkish Data. He reached 85 percent success rate if data consider positive or negative

2.2. BACKGROUND

2.2.1. TWITTER

Twitter, for this very moment now, is the most used micro-blogging in the world. Everyone in the world, no matter what they are interested in nor what they think about anything can share them as long as they are within 140 characters. At the same time other people can share those tweets as they are written (this is called retweeting!). Any kind of pictures, video or similar documents can be shared. The words or thoughts marked with # are called hashtag and it indicates that word or thought is wanted to be put forward.

The twitter data used in this thesis are collected by sniffing -thanks to API's of Twitter and Java Programming Language- all the tweets which were tweeted about some specific brands.

2.2.2. R PROGRAMMING LANGUAGE

R, a statistical calculator and software, is also a programming language. R, having been found by Ross Ihaka and Robert Gentleman from Auckland University in New Zealand, is still being developed R Development Team. R is also called "GNU S" due to being open source version of S programming language. It is inevitable not to use R when data gathering, processing and statistics are in questions. It has a huge potential to be one of the most used programming languages and tools of developing big data world.

3. METHODOLOGIES

This research is examining the ability of different machine learning classifiers in classifying sentiment in twitter feeds. Five common machine learning methods for supervised classification of data include support vector machines, random forests, boosting, maximum entropy, and artificial neural networks. These methods have demonstrated ability to classify data including text data with desirable performance outputs across a wide range of problem types (Caruana et al., 2008).

3.1. DATA GATHERING AND PREPROCESSING

Twitter feeds from hepsiburada.com and pegasus.com were obtained. The 717 tweets from hepsiburada.com and 129 tweets from pegasus.com (846 total tweets) consisted of opinions about travel. The language of the twitter feeds was Turkish. In addition, a lexicon of 3579 words was generated with sentiment ranging from very negative (-5) to very positive (+5), including a variety of sentiments in-between. The lexicon ratings were performed by the author. Of these words in the lexicon, 2,821 had a score of zero, not having any positive nor negative sentiment.

The twitter feeds were then pre-processed using R Statistical Software (version 3.2.3) with packages for text-mining (tm, version 0.6-2) and text classification (RTextTools, version 1.4.2). A document-term matrix was created, excluding words with less than two letters, removing numbers, removing punctuation, and converting punctuation to lower case. The matrix consisted of 3,732 words (listed as columns) across the 846 tweets (listed as rows) with number of occurrences in the tweet as values. Most words were represented in a tweet just once with 138 tweets included a word twice and 3 tweets included a word three times.

An example of a document-term matrix based on three tweets is described in Table 1. The tweets are “mobil alisveris yukseliste..” (document 1), “umarim yarin teslimatim yapilir..” (document 2), and “resmen kandirildim cok yazik” (document 3). Each tweet is considered a “document” and are represented by rows. Each word is considered a “term” and represented in the column with each number representing a word ordered alphabetically (e.g., “alisveris” = 1, “cok” = 2, “kandirildim” = 3, “mobil” = 4, etc.). The cells represent the number of times a term (i.e., a word) is mentioned in the document (i.e., a tweet). The document term matrix is a sparse matrix, with many cells containing no information. If not empty, most of the cells in the table are represented by the value of 1, which is the number of times that a term is represented in a document.

Table 3.1: Sample document term matrix

Document	1	2	3	4	5	6	7	8	9	10	11
1	1			1							1
2						1	1	1	1		
3		1	1		1					1	

3.2. SCORING OF SENTIMENT OF TWEETS

For each tweet, a total sentiment score was calculated based upon the matrix and the lexicon. The total sentiment score for a tweet was equal to the sum of the sentiment score of each word times the number of the times the word was represented (Equation 3.1).

$$S_{tweet} = \sum(s_n w_n) \qquad \textbf{(3.1)}$$

Stweet = total sentiment score for a tweet

sn = sentiment score of a word n

wn = number of times word n appears in a tweet

Equation 3.1 could be applied to Table 1. As mentioned previously, most words had a sentiment score of 0. For the tweets used to create Table 1, “umarim” (Term 7) had a score of +1, “cok” (Term 2) had a score of +1, and “kandirildim” (Term 3) had a score of -4. The total sentiment score for document 1 would be 0 (0*1 + 0*1 + 0*1), for document 2 would be 1 (1*1 + 0*1 + 0*1 + 0*1), and for document 3 would be -3 (1*1 + -4*1 + 0*1 + 0*1). From the Table 1 example, documents 1, 2, and 3 would have raw scores of 0, 1, and -3, respectively.

Under the scoring system from Equation 3.1, the sentiment scores for a tweet ranged from -11 to +11 (Table 2). The raw scores were further processed to develop two alternative scores. A tweet score was converted to a simple positive/negative scoring system (-1, 0, +1) based on the total positive, neutral, and negative scores. Tweets with a total score of -10 or -3 were both considered negative tweets and the magnitude of the negative or positive score was ignored. Also, a tweet score was converted to a scaled score (-2, -1, 0, +1, +2) whereby scores of -11 to -5 were reassigned a score of -2, scores of -4 to -1 were reassigned a score of -1, scores of 1 to 4 were reassigned as score of 1, and scores of 5 to 11 were reassigned a score of 2. This scoring differentiated the very negative or very positive scores. The simple positive/negative scoring system for documents 1, 2, and 3 would be 0, 1, and -1, respectively. The scaled scores for documents 1, 2, and 3 would also be 0, 1, and -1, respectively.

These two processed scores (positive/negative and scaled) were used as the response variable for the supervised classification step. The 846 tweets were reshuffled to create a random ordering of tweets for further analysis. The tweets were separated into a class for training the models (700) and for testing (146).

Table 3.2: Distribution of sentiment scores for tweets

Score	Number of Tweets
-11	1
-9	2
-8	5
-7	11
-6	9
-5	39
-4	34
-3	55
-2	72
-1	136
0	229
1	107
2	64
3	31
4	21
5	17
6	7
7	1
8	2
9	1
10	1
11	1

The model training and testing were performed using the RTextTools package (Jurka et al. 2013). Within the RTextTools package, classification algorithms used for the analysis included support vector machines (SVM), random forests, boosting, maximum entropy (MAXENT), and artificial neural networks (ANN). Models were generated using the 700 tweets for model training and the predictions based on the 146 tweets for testing were compared with the actual sentiment values.

For all final model outputs the proportion accuracy (Equation 3.2), precision (Equation 3.3), and recall (Equation 3.4) were calculated (Powers, 2011).

$$\text{Accuracy} = (\Sigma\ \text{True Positive} + \Sigma\ \text{True Negative}) / \Sigma\ \text{Total Population} \qquad \textbf{(3.2)}$$

$$\text{Precision} = \Sigma\ \text{True Positive} / \Sigma\ \text{True Positive} + \Sigma\ \text{False Positive} \qquad \textbf{(3.3)}$$

$$\text{Recall} = \Sigma\ \text{True Positive} / \Sigma\ \text{True Positive} + \Sigma\ \text{False Negative} \qquad \textbf{(3.4)}$$

In addition, receiver operating characteristic (ROC) curves were determined u using the pROC package (version 1.8) within R Statistical Software. Within the ROC curve, the x-axis values represent (1 – specificity (Equation 3.5) (Metz, CE (1978)). The y-axis values represent sensitivity, which is the same as recall (Equation 3.4)

$$\text{Specificity} = \Sigma\ \text{True Negative} / \ \Sigma\ \text{True Negative} + \Sigma\ \text{False Positive} \qquad \textbf{(3.5)}$$

3.3. CLASSIFICATION USING SUPPORT VECTOR MACHINES

SVMs are supervised learning algorithms that use hyperplanes for classifying data (Boser et al., 1992; Vapnik, 1998; Karatzoglou et al., 2006; Hastie et al. 2009). The hyperplanes are constructed by finding the margins that separate the closest points of different classes (Figure 3.1). The points lying along the boundaries representing the support vectors. For simple problems, a simple hyperplane can separate the classes readily. For more complicated problems, a hyperplane does not result in complete separate with points on the wrong side of the margin (Figure 3.2). The hyperplanes are defined by

$$h(x) = \ w^T x \qquad \textbf{(3.6)}$$

where **w** is a weight vector (i.e., the support vector) with points on a hyperplane being along h(x) = 0.

For a two-class classification problem, for a training data set (x_i, y_i) where $x_i \in \mathbf{R}$ and $y \in \{1,-1\}$, the support vector machines optimize:

$$\begin{aligned} \min_{w,\varepsilon} \quad & \frac{1}{2} w^T w + C \textstyle\sum_i \ \varepsilon_i, \\ subject\ to \quad & y_i(w^T \phi(x_i) \geq 1 - \ \varepsilon_i, \\ & \varepsilon_i \ \geq 0. \end{aligned} \tag{3.7}$$

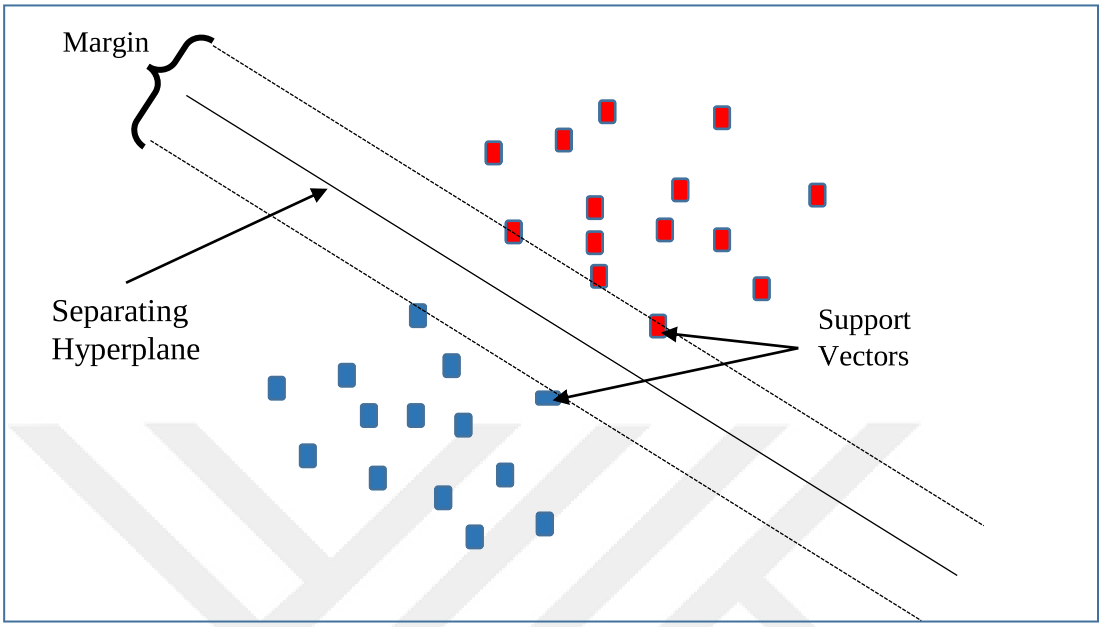
where C is a cost factor that controls for the hardness/softness of the margins, ε is a distance of a data point on the wrong side of a margin from its expected margin, and ϕ is a high dimensional mapping of the input data as defined by a kernel function. For ϕ, a common kernel function for separating with a non-linear hyperplane is the radial basis function:

$$k(x, x') = exp(e^{-\gamma \|x - x'\|^2}) \tag{3.8}$$

Where γ is a kernel parameter for the radial basis function.

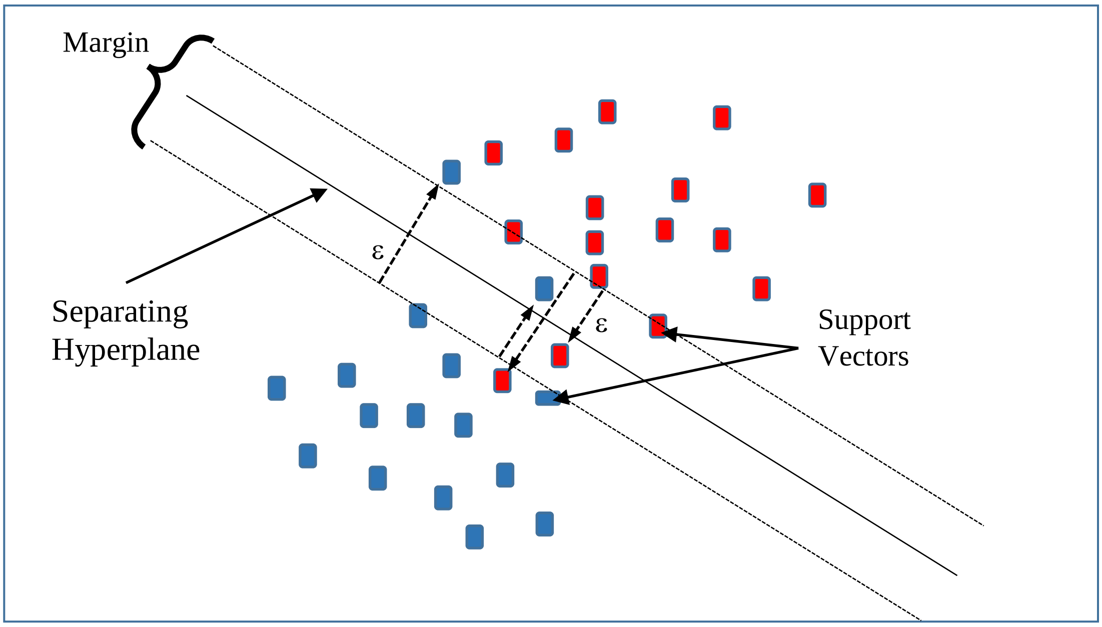
Classification is based off of the ability of the support vector machines to separate high dimensional space into specific regions. Membership within a region determines the final classification of the data point.

Figure 3.1: Support vector machine hyperplane and margins for separate case with points on the wrong side of the margin showing the distance (ε) from the margin



Resource: Boser et al., 1992; Vapnik, 1998; Karatzoglou et al., 2006; Hastie et al. 2009

Figure 3.2: Support vector machine hyperplane and margins for nonseparable (overlap) case with points on the wrong side of the margin showing the distance (ε) from the margin



Resource: Boser et al., 1992; Vapnik, 1998; Karatzoglou et al., 2006; Hastie et al. 2009.

Within the R Statistical Software, the RTextTools package was used. The SVM features of the RTextTools package relied on the e1071 package. The e1071 package was based off of the libsvm C++ implementation. Within the RTextTools implementation of the e1071 package, parameters for cost (C) and gamma (γ) need to be adjusted for to maximize separation for the classification analysis. The cost parameter (C) controls for the cost of misclassification of the SVMs. This effectively controls for the rigidity of the boundary with large C-values resulting in very distinct boundaries (hard margins) and small C-values allowing for diffuse boundaries (soft margins) with more misclassifications along the margins. Consequently, large C-values give low bias but high variance, whereas small C-values give high bias, but low variance. The gamma parameter (γ) controls for the shape of higher dimension separators with large γ-values resulting in softer, broader peaks and small γ-values giving pointed higher dimensional peaks. Similarly, large γ-values would give high bias, low variance, whereas small γ-values would give low bias, high variance.

The best performing SVM models using positive/negative sentiment scores and the scaled sentiment scores were determined by adjusting parameters using the training data to develop models and the test data to evaluate the models. Varying C-values (1000, 100, 10, 1, and 0.1) and γ-values (0.00023, 0.00024, 0.00025, 0.00026, 0.00027, 0.00028, 0.00029, and 0.00030) were examined to fine-tune the parameters. For the SVM models using positive/negative outputs, a C-value of 1000 and a γ-value of 0.00028 resulted in the best performance. For the SVM models using the scaled scores, a C-value of 100 and a γ-value of 0.00028.

3.4. CLASSIFICATION USING RANDOM FORESTS

The random forest (RF) algorithm is a supervised learning algorithm using an ensemble of models that collectively vote on a response variable (Breiman, 2001; Liaw and Wiener, 2002). A model within the ensemble is a single multiple classification and regression tree (CART), an algorithm that models the relationship between a response variable and a set of explanatory variables through a series of partitioning rules such that classification errors from each split is minimized (Breiman et al., 1984). A series of splits would eventually result in classification of all the data points. In each split, the frequency of correctly classifying a data point (f_i) or not (1-f_i) could be calculated and the overall level of classification performance would be based on reducing the Gini impurity. Gini impurity is computed as:

$$I_G = \sum_i^n f_i(1 - f_i) \qquad \textbf{(3.9)}$$

The two notable differences between CARTs and a RF algorithm are the generation of many trees for RFs (default of 500) based on a bootstrapped sample of the original data (as opposed to all the data points) and the use of only a subset of explanatory variables (as opposed to all of the variables) at each split. During bootstrapping for a single tree, about two-thirds of the data points are used for model development. At each split, the square root of the number of variables is used at each split. The use of both bootstrapping and using a subset of the explanatory variables insures that the trees are not correlated with one another. An individual tree has a tendency to be highly sensitive to the noise in the dataset, resulting in overfitting and poorer algorithm performance and also poorer stability with any changes in data sets. With the use of multiple uncorrelated trees in RF, the noise and the variance is reduced, resulting in better algorithm performance.

The entire RF algorithm uses all the trees within the random forest group to classify data. Each tree represents a single vote on classification and the majority vote of all the trees would determine the class of a data point.

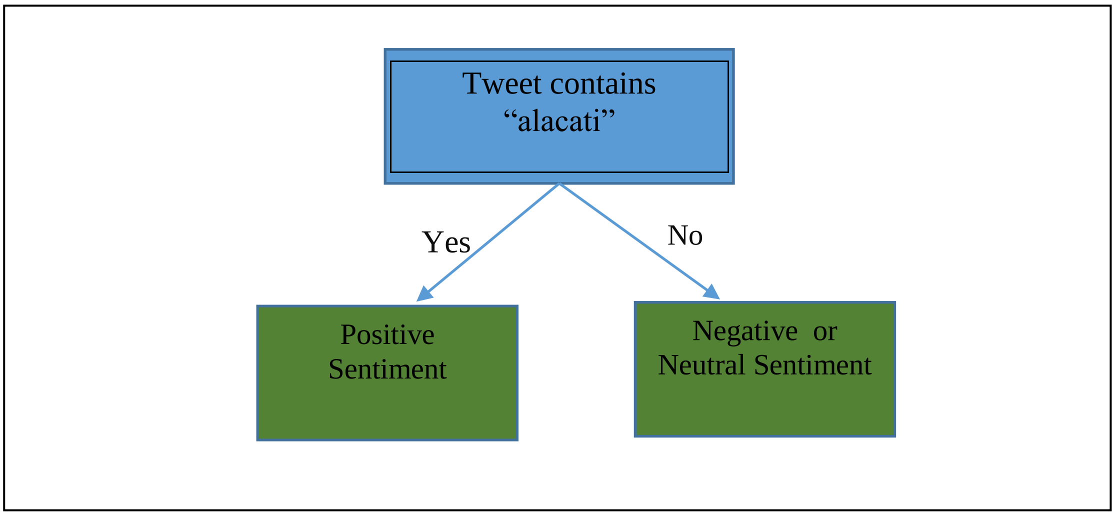
Within the R Statistical Software, the RTextTools package was used. The RF features within the RTextTools package relied on the randomForest package was used. The randomForest algorithm is based off of the original Fortran program for random forests. Within the RTextTools implementation of the randomForest package, the main parameter subject to control is the number of trees generated. The tradeoff in selecting the number of trees is between speed of processing and potential overfitting.

The best performing RF models using positive/negative sentiment scores and the scaled sentiment scores were determined by adjusting the parameter for number of trees using the training data to develop models and the test data to evaluate the models. Varying the number of trees (125, 250, 500, and 750) was examined to fine-tune the parameters. For the RF models using positive/negative outputs, a 250 trees resulted in the best performance. For the SF models using the scaled scores, 250 trees resulted in the best performance.

3.5. CLASSIFICATION USING BOOSTING

Boosting is a type of supervised learning algorithm that uses many weak classifiers to create an accurate combined classifier (Freud and Shapire, 1996; Dettling and Bühlmann, 2003). The multi-stage algorithm gives greater weight to misclassified data with each iteration. Elements of the algorithm include the type of base classifier, the weighting, and the number of iterations for the entire process. The base classifier used is often a classification and regression tree 'stump,' which consists of a tree with just two terminal nodes (Figure 3.3). The weights are determined by the algorithm at each stage. The user inputs the maximum number of iterations.

Figure 3.3: Example of a decision tree stump.



The algorithm results in multiple iterations with the final classification determined by a weighted majority of the weak classifiers. For each iteration, greater weight is given to misclassified data points and lesser weight is given to correctly classified data points. The overall algorithm consisted of a weighted majority vote of the classifiers. In the multi-stage process for binary classification, the initial settings are set for the committee function ($F^{(0)}(x) = 0$) and for initial probabilities $p^{(0)}(x) = ½$, which is the probability of y equaling 1 for a given explanatory variable. To fit the weak learner, for iterations ranging from m = 1 to the final number of iterations (M) and for data points ranging from i = 1 to the total number of data points (n), the algorithm fits the classifiers to compute weights (w) and an intermediate response (z):

$$w_i^m = p^{(m-1)}x_i \cdot \left(1 - p^{(m-1)}x_i\right), \qquad \textbf{(3.10)}$$

$$z_i^m = \frac{y_i - p^{(m-1)}x_i}{w_i^m}$$

which would then be used to fit the regression stump in Figure 3.3 by weighted least squares:

$$f^m = argmin \sum_i \; w_i^m \left(z_i^m - f(x_i)\right)^2 \qquad \textbf{(3.11)}$$

In the process, the committee function and the probabilities are updated, feeding into a vote of the classifiers:

$$F^m(x_i) = F^{(m-1)}(x_i) + \frac{1}{2} f^m(x_i), \qquad \textbf{(3.12)}$$

$$C^m(x_i) = sign(F^m(x_i)),$$

$$p^m(x_i) = \left(1 + exp(-2 \cdot F^m(x_i)\right)^{-1}$$

The committee function is an estimate of a the half of a log-odds ratio, which provides the basis of the name of the algorithm (logitboost):

$$F(x) = \frac{1}{2} log\left(\frac{p(x)}{1-p(x)}\right) \qquad \textbf{(3.13)}$$

The algorithm also converts multiclass datasets into a binary dataset. For a dataset with a 3-class response variable, the algorithm would divide the dataset into class C and class A/B, and then further divide class A/B into class A and class B.

Within the R Statistical Software, the RTextTools package was used. The boosting features of the RTextTools package relied on the caTools package. The caTools package was based off of the LogitBoost package and uses the rpart package, which allows for the use of the classification and regression tree algorithm. Within the RTextTools implementation of caTools, the parameter for maximum iterations controls the development of the algorithm. The number of iterations controls how long the boosting process occurs. With increased number of iterations, there could be the potential for overfitting of the model. Controlling the number of iterations would insure that model fits the data without overfitting.

The best performing boosting models using positive/negative sentiment scores and the scaled sentiment scores were determined by adjusting parameters using the training data to develop models and the test data to evaluate the models. Varying the number of iterations (25, 50, 75, 100, and 125) was examined to fine-tune the parameters. For the boosting models using positive/negative outputs, 50 iterations resulted in the best performance. For the boosting models using the scaled scores, 75 iterations resulted in the best performance.

3.6. CLASSIFICATION USING MAXIMUM ENTROPY

Maximum entropy (MAXENT) is a probability distribution estimation algorithm that presupposes uniform distributions of data (Nigam et al. 1999; Jurka, 2012). When the data classes are uniform, entropy is maximized. With the MAXENT algorithm, training data outline a set of constraints, represented as expected classification for a set of data points.

For text data, the algorithm estimates conditional distribution based on the presence of specific words. The presence of specific words would increase the probability of a certain classification compared to an equal distribution had the specific word not have been present. As an example, in a three category classification of positive, neutral, and negative tweets, if past observations indicated that the 60 percent of the tweets containing the word “ugly” are in the “negative” class, one could infer that seeing the word “ugly” in a tweet would correspond to a 60 percent chance that tweet is negative,” and 20 percent chance it would be positive or 20 percent chance it would be neutral. Without that word, one would conclude that there would be about 33 percent chance the tweet would be in any of the three classes. The presence of certain words would increase the likelihood of categorization compared to maximum entropy null state. In reality, most situations would be more complicated, but the same principles would apply.

For text classification, MAXENT based on training data would allow for classification of new data. For a document (d) and a class (c) for i = 1 to a total number of documents (n), a conditional distribution could be created to restrict testing data to have the same expected value as the training data:

$$\frac{1}{n}\sum_i \; f_i(d_i,c_i) = \sum_i \; P(d_i)\sum_i \; P(c|d)f_i(d_i,c_i) \qquad \textbf{(3.14)}$$

Within the R Statistical Software, the RTextTools package was used. The MAXENT features of the RTextTools package relied on the maxent package. Within the RTextTools implementation of maxent, the parameters for L2 regularization and the number of documents that are held out control the performance of the algorithm.

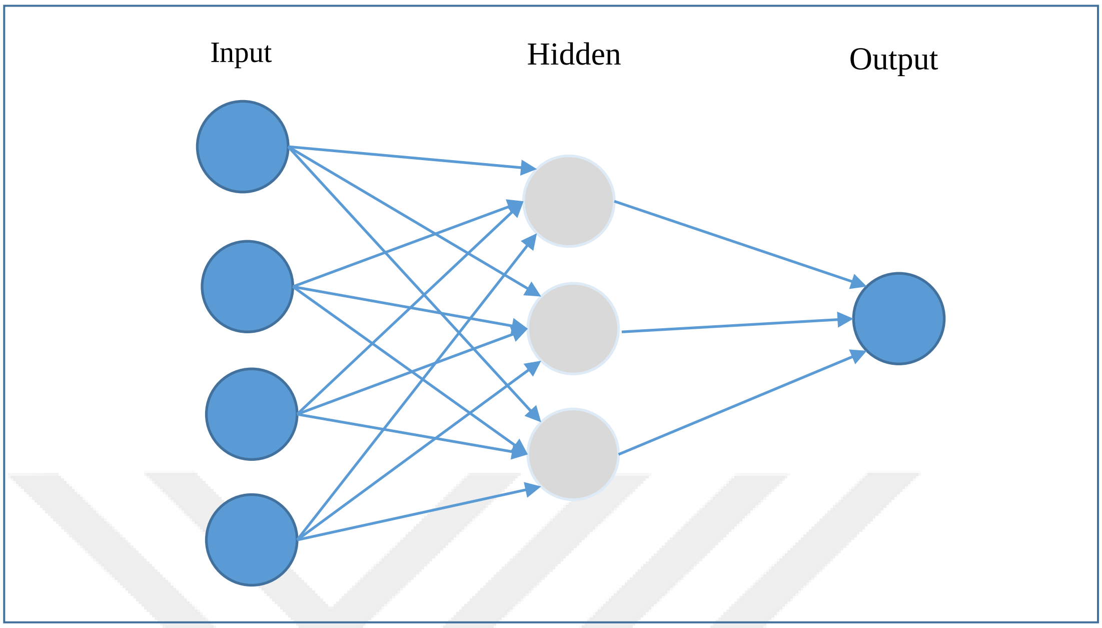
The best performing MAXENT models using positive/negative sentiment scores and the scaled sentiment scores were determined by adjusting parameters using the training data to develop models and the test data to evaluate the models. Varying the parameter for L2 regularization (0, 0.2, 0.4, 0.6, 0.8, and 1) was examined to fine-tune the parameters. For the MAXENT models using positive/negative outputs, a L2 regularization value of 0.2 resulted in the best performance. For the MAXENT models using the scaled scores, a L2 regularization value of 0.2 resulted in the best performance.

3.7. CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

Feed forward artificial neural network (ANN) is a type of supervised learning algorithm that simulates neural systems (Jain et al., 1996; Venables and Ripley, 2002). In the system, connected artificial neurons exchange messages with one another with the connections having numeric weights that can be tuned with experience. Training data could be inputed into an ANN model to tune the model for the purposes of classification.

Figure 3.4 shows a feed forward artificial neutral network with a hidden layer. Each of the input neurons interact with the environment receiving information directly. The hidden neurons do not interact with the environment, but collect information from the input neurons. The hidden neurons deliver a weighted sum of signals, which results in an output if the sum of signals exceeds a threshold.

Figure 3.4: A single layer feed forward ANN with 4 units in the hidden layer



The input and hidden neurons deliver information that follow specific criteria. The inputs that are delivered from input to hidden and from hidden to output neurons may be excitatory or inhibitory. The sum of the all the weighted excitatory and inhibitory neurons determine the output based on the inputs (x), and weights (w) in light of the threshold (u) and the step function that processes the signals θ:

$$y = \theta \sum_i w_i x_i - u \qquad \textbf{(3.15)}$$

The ANN algorithm is iteratively developed from an initial state. The weights and thresholds are given small random numbers. The algorithm would then take the training data and develop an output from the explanatory variables. The weights would then be updated for the iteration number (t) according to the following equation based on the desired/predicted output (d), the actual output (y), the input (x), and the step size (η):

$$w_i(t+1) = w_i(t) + \eta(d-y)x_i \qquad \textbf{(3.16)}$$

In using the ANN algorithm, a few input parameters are needed. The number of units in the hidden layer needs to be specified. The maximum number of iterations needs to be chosen, with large values having a risk of overfitting. To counter the potential of overfitting, decay values for the weights could be specified to control overfitting.

Within the R Statistical Software, the RTextTools package was used. The neural network features of the RTextTools package relied on the nnet package. Within the RTextTools implementation of nnet, the parameters for size, decay, and maximum iterations control the performance of the netural network.

The best performing neural network models using positive/negative sentiment scores and the scaled sentiment scores were determined by adjusting parameters using the training data to develop models and the test data to evaluate the models. Varying the parameter for size (1, 2, 3, and 4), decay (5 x 10^{-2}, 5 x 10^{-3}, 5 x 10^{-4}, and 5 x 10^{-5}), and maximum iterations (100, 150, 200, 250, and 300) were examined to fine-tune the parameters. For the neural network models using positive/negative outputs, the combination of parameters that had the best performance was a size of 3, a decay value of 5 x 10^{-5}, and maximum iteration value of 150. For the neural network models using the scaled scores, the combination of parameters that had the best performance was a size of 2, a decay value of 5 x 10^{-2}, and a maximum iteration value of 300.

4. RESULTS

4.1. TWITTER DATA

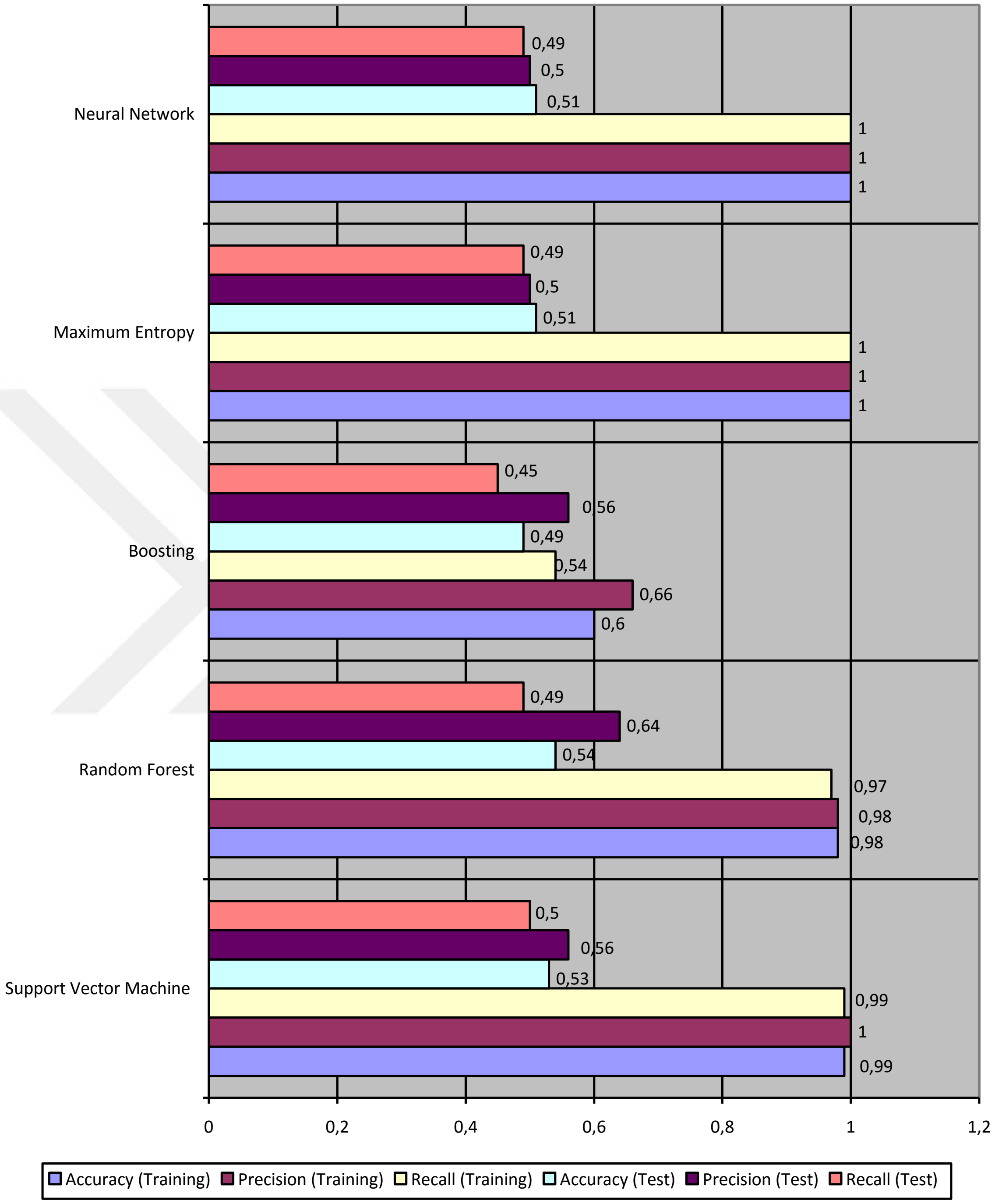
Overall, the twitter data had a higher degree of negativity. The training data had tweets that were 43 percent negative, 26 percent neutral, and 31 percent positive. On a five-category scaled breakdown, the tweets were 9 percent very negative, 32 percent negative, 33 percent neutral, 23 percent positive, and 3 percent very positive. The test data had tweets that were 41 percent negative, 33 percent neutral, and 26 percent positive. On a five-category scaled breakdown, the tweets were 8 percent very negative, 36 percent negative, 26 percent neutral, 27 percent positive, and 4 percent very positive.

4.2. ACCURACY ASSESSMENT

The examination of sentiment using five different algorithms resulted in very similar results for the analysis of positive/negative sentiment scores and for the scaled sentiment scores. The accuracy based on the training data set was higher than the accuracy determined from the test data set for both the positive/negative sentiment scores (Table 4.1) and the scaled sentiment scores (Table 4.2). With support vector machines, random forests, maximum entropy, and artificial neural networks, the difference between the training and test data accuracy was over 0.4, whereas with boosting, the difference was no greater than 0.25.

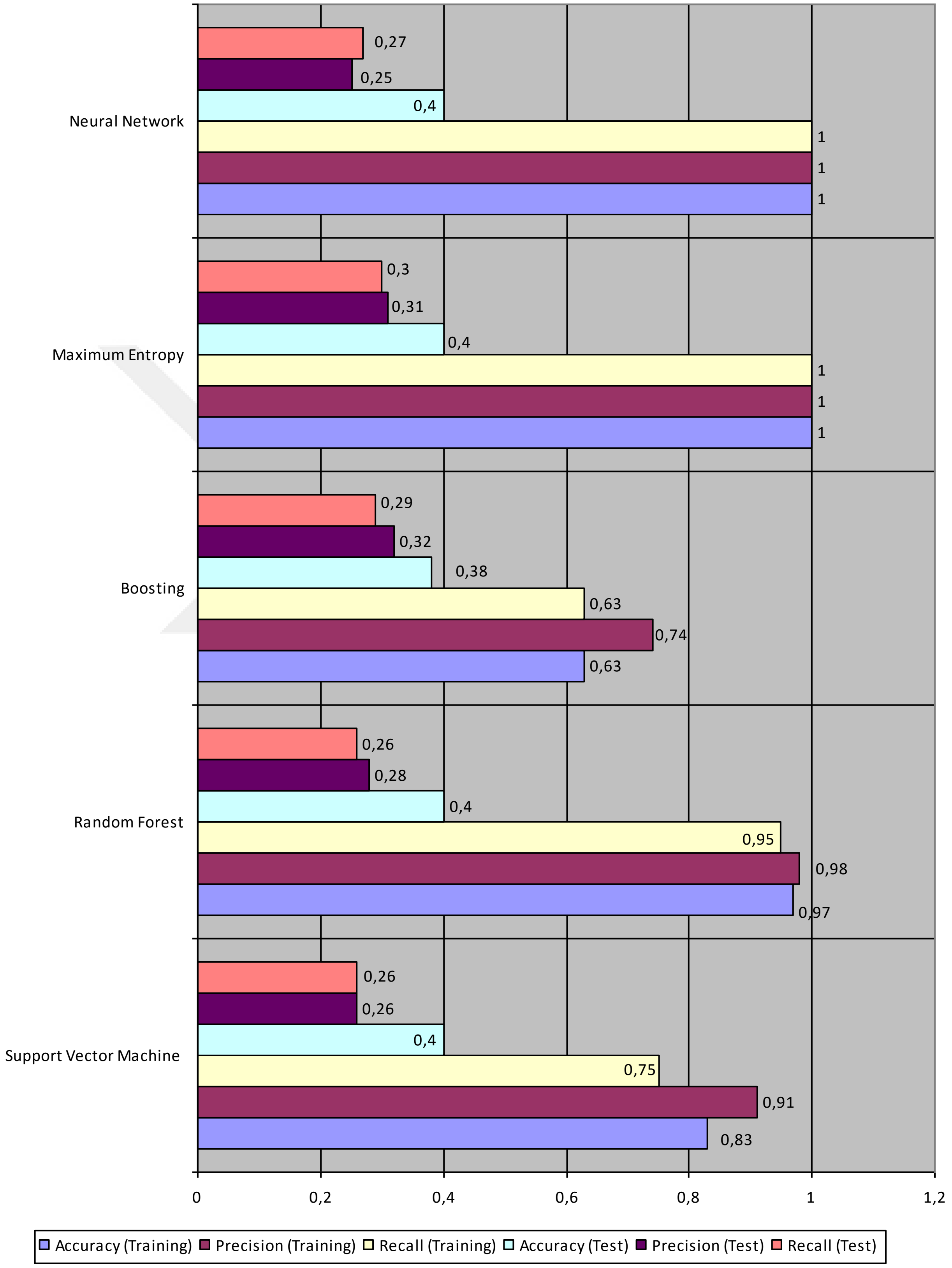
For the accuracy assessment of the positive/negative sentiment scores (Table 4.1), the values ranged from 0.49 to 0.54. The precision and recall values were similar. The most accurate classification algorithm was the random forest algorithm at 54 percent. The least accurate classification algorithm was the boosting algorithm.

Table 4.1: Accuracy assessment of the positive/negative sentiment scores



For the accuracy assessment of the scaled sentiment scores (Table 4.2), the values ranged from 0.38 to 0.40. The precision and recall values were similar. The classification algorithm for the scaled sentiment scores had lower accuracy values (10-15 percent less) compared to the positive/negative sentiment score classification. The most accurate classification algorithms were support vector machine, random forest, and the maximum entropy algorithms. The least accurate classification algorithm was the boosting algorithm.

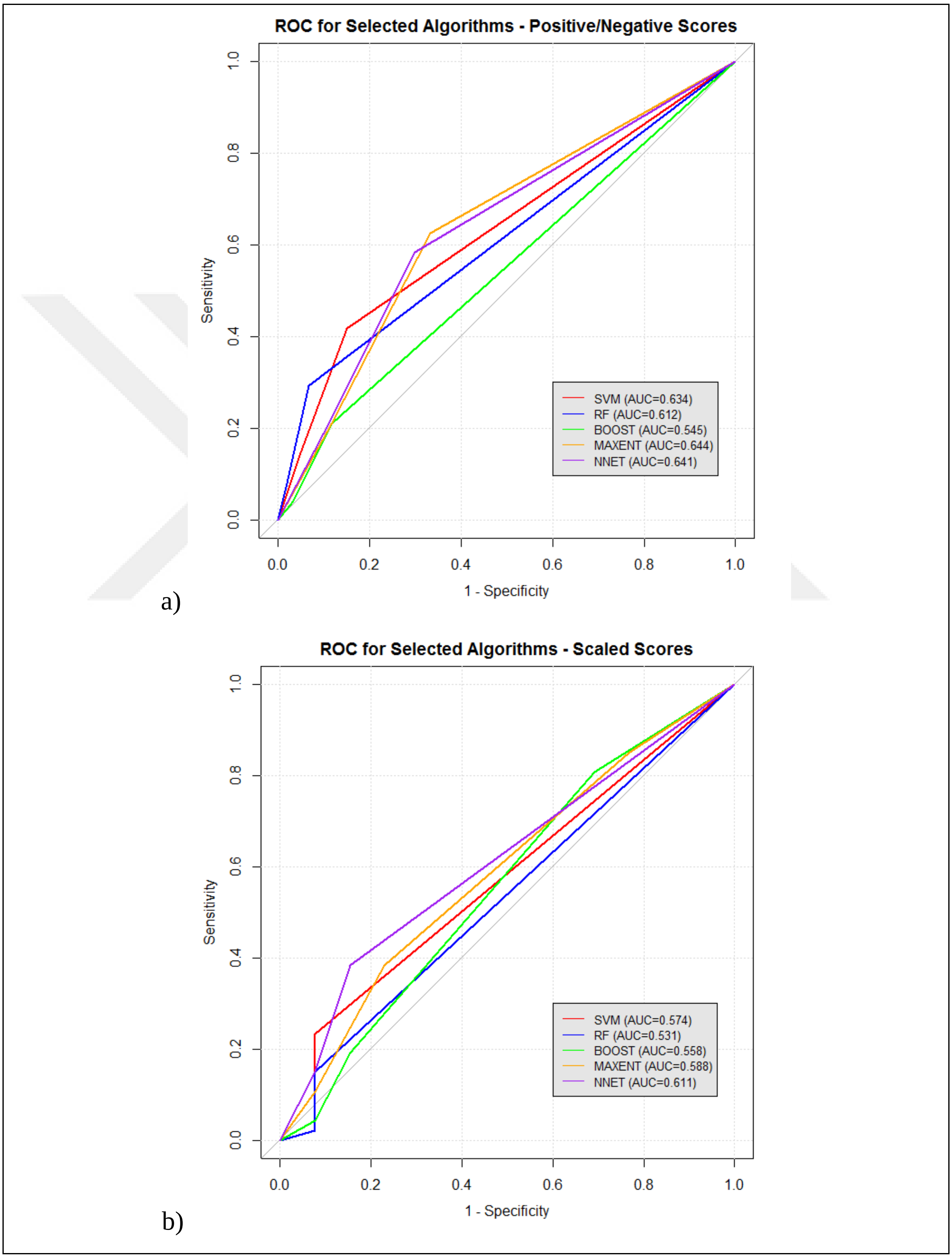
Table 4.2: Accuracy assessment of the scaled sentiment scores



4.3. RECEIVER OPERATING CHARACTERISTIC CURVE

Figures 4.1 show the ROC curves for each of the five classification algorithms for the positive/negative scores and for the scaled scores. The area under the curve for the positive/negative scores ranged from 0.545 to 0.644. The boosting algorithm provided the lowest AUC and the maximum entropy provided the highest AUC. The area under the curve for the scaled scores ranged from 0.531 to 0.588. The random forest algorithm provided the lowest AUC and the maximum entropy provided the highest AUC.

Figure 4.1. : Receiver operating curves for classification of twitter sentiments using support vector machines, random forests, boosting, maximum entropy, and artificial neural networks for a) positive/negative scores and b) scaled scores



5. DISCUSSION

5.1. OVERFITTING

The examination of accuracy from training and test data indicated the algorithms had a large degree of overfitting. This was especially true for support vector machines, random forests, maximum entropy, and artificial neural networks. Boosting had a lesser degree of overfitting. These results reinforce the need to have accuracy assessments on separate data from the data used to develop the model in order to avoid over-optimistic assessments of accuracy.

5.2. ACCURACY

The level of accuracy declined with the increase in number of classes. The positive/negative score classification involved three classes (positive, neutral, and negative) and the scaled score classification involve give classes (very positive, positive, neutral, negative, and very negative). As more classes are added, the spatial extent of the classes were reduced, resulting in a smaller classification target footprint. The consequence would be greater likelihood of misclassifications.

The percent accuracy for all algorithms were similar and around 0.5-0.55 for the positive/negative scores and the 0.4 for the scaled scores. These accuracy scores were similar to what was observed before for other studies classifying text using machine learning algorithms. Previously, Hsu et al. (2010) recorded 47 percent accuracy for a five-category sentiment analysis project using SVMs. Similarly, Socher (2014) recorded 49.7 percent accuracy for the same five-category sentiment analysis. Also, Go et al. (2009) recorded 80.5 percent and 82.2 percent accuracy for classifying positive and negative (no neutral) sentiments using maximum entropy and SVMs, respectively. In addition, Pang et al. (2002) recorded accuracies of 72.8 percent to 82.9 percent using support vector machines on positive/negative sentiments (no neutral sentiments).

The main distinction between the higher accuracy measurements in past sentiment analyses with machine learning algorithms with the current effort is the number of classes. The studies recording accuracies in the 80 percent range dealt with only positive and negative comments (two classes) and did not consider neutral comments. The current effort include neutral comments. Thus, it was understandable why the accuracies were not as great with the current effort. The other studies that looked at five classes (Hsu et al. 2010, Socher 2014) also reported less accurate assessments.

In light of the results from other studies and in consideration of the differences, the accuracy of the classifications from the current effort were within reason. With three classes, about 50-55 percent accuracy would be appropriate. After factoring in the advanced work performed to fine-tune the parameters, the level of accuracy for classification of sentiment using this dataset was high highest level expected.

In the light of these datas, that SVM, Random Forest and Maximum Entropy Algorithms has got such a high potential is quite remarkable. It seems that if especially tendency towards Maximum Entropy Algorithm is raised, it is very likely to achieve very succesfull works.

In addition, when looking at literature, there is no comparison especially on Turkish Language with Support Vector Machines, Random Forests, Boosting, Maximum Entropy, and Artificial Neural Networks classification algorithms thru R Programming Language. Therefore, taking this fact into consideration, current work is first of its kind in that field and apt to be developed further. Having said this can be claimed that it stands in midts of technical literature' results due to its number of classes.

5.3. ALGORITHMS

Another phenomenon was the convergence in performance among the five algorithms. The differences among algorithms were relatively small. Other comparative studies have shown that different machine learning algorithms have had markedly different results (Caruana and Niculescu-Mizil, 2006; Caruana et al., 2008). The current effort did show differences in performance, but the differences suggest similar performance.

There are natural questions on whether the five algorithms used were the best algorithms. These five represent the highest performing machine learning algorithms (Caruana and Niculescu-Mizil, 2006; Caruana et al., 2008). It is unlikely that another algorithm would have drastically greater level of performance for the given data set.

The algorithms used within this study were controlled through commands within the RTextTools package. The commands within RTextTools facilitated pre-processing of the twitter feeds, removing two-letter words, punctuation, and numbers. The commands within RTextTools called upon machine learning algorithms in existing R packages, facilitating the use of multiple machine learning packages within a systematic approach.

However, within R Statistical Software, alternate packages not called upon by RTextTools may result in different performances. For artificial neural networks, the neuralnet package may offer some improvements. For boosting, the gbm package may offer greater fine-tuning features and improvements. For random forests, the conditional inference forests of the party package may offer improvements over the known biases of the randomForest package.

5.4. LEXICON

A methodological issue may have been the scoring of the lexicon. The lexicon sentiment scores were assigned by one individual. Without having an established lexicon with associated emotion and sentiment values[1], the assignment of sentiment scores by just one individual was needed. Due to this methodological limitation, there may have been a bias. If the lexicon was developed by a larger group of people or have been peer-reviewed, the scoring may have been different. This may have resulted in a more clearly defined, consistent scoring system that would have led to clearer classifications.

[1] mpqa.cs.pitt.edu/ provides a list of English words and associated subjectivity values.

6. CONCLUSIONS AND FUTURE WORK

The current work classified twitters very positive, positive, neutral, negative, and very negative sentiments with estimated accuracies of up to 55 percent for three-class outputs using five machine learning algorithms including Support Vector Machines, Random Forests, Boosting, Maximum Entropy, and Artificial Neural Networks. The outputs were similar across the different algorithms. The current work advances sentiment analysis work using a comparative evaluative approach.

Future work would involve ways to improve the outputs. The current work involved five machine learning algorithms. Whether these are the best algorithms needs further consideration in light of alternative versions of a particular algorithm. Future work could also use additional machine learning algorithms. In light of the comprehensive approach undertaken with the current work, which includes a diversity of algorithms and thorough efforts to fine-tune the algorithms, drastic improvements may be unlikely.

REFERENCES

Periodical Publications

Breiman, L. 2001. “Random forests,” *Machine Learning*, 45, pp. 5-32.

Dettling, M., and Bühlmann, P. 2003. “Boosting for tumor classification with gene expression data,” *Bioinformatics*, 19(9), pp. 1061-1069.

Jain, A.K., Mao, J., and Mohiuddin, K.M. 1996. “Artificial neural networks: a tutorial,” *Computer*, 3, pp. 31-44.

Jurka, T.P. 2012. “maxent: An R package for low-memory multinomial logistic regression with support for semi-automated text classification,” *The R Journal*, 4(1), pp. 56-59.

Jurka, T.P., Collingwood, L., Boydstun, A.E., Grossman, E., and van Atteveldt, W. 2013. “RTextTools: a supervised learning package for text classification,” *The R Journal*, 5(1), pp. 6-12.

Karatzoglou, A., Meyer, D., and Hornik, K. 2006. “Support vector machines in R,” *Journal of Statistical Software*, 15(9), pp. 1-28.

Liaw, A., and Wiener, M. 2002. “Classification and regression by randomForest,” *R News*, 2/3, pp. 18-22.

Metz, CE (1978). "Basic principles of ROC analysis". *Semin Nucl Med. 8* (4): 283–98.

Powers, David M W (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". *Journal of Machine Learning Technologies*, 2, pp. 37–63.

Other Publications

Arora D, Kin Fun Li, and Neville S. 2015. *“Consumers’ Sentiment Analysis of Popular Phone Brands and Operating System Preference Using Twitter Data: A Feasibility Study.”* 2015 IEEE 29th International Conference on Advanced Information Networking and Applications.

Bhuta, S., 2014. *“A Review of Techniques for Sentiment Analysis of Twitter Data.”* IEEE Computer Society.

Boser, B., Guyon, I., and Vapnik, V. 1992. “A training algorithm for optimal margin classifier,” in *Proc. 5th Annual ACM Workshop Computational Learning Theory*, pp. 144–52.

Breiman, L., Friedman, J.H., Oshen, R.A., and Stone, C.J. 1984. *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books and Software, Monterey, CA.

Caruana, R., Karampatziakis, N., and Yessenalina, A. 2008. “An Empirical Evaluation of Supervised Learning in High Dimensions,” in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland.

Caruana, R., and Niculescu-Mizil, A. 2006. “An Empirical Comparison of Supervised Learning Algorithms,” in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA.

Coletta, L. F. S., Hruschka E. R., Silva N. F. F. 2014. *“Combining Classification and Clustering for Tweet Sentiment Analysis.”* 2014 Brazilian Conference on Intelligent Systems.

Demirci, S. 2014. *“Emotion Analysis on Turkish Tweets.”* Thesis Submitted to Graduate School Of Natural and Applied Sciences of Middle East Technical University.

Duwairi M.R. 2015. *“Sentiment Analysis for Dialectical Arabic.”* 2015 6th International Conference on Information and Communication Systems (ICICS).

El-Beltagy, S.R. 2013. *“Open Issues in the Sentiment Analysis of Arabic Social Media: A Case Study.”* IEEE Computer Society.

Erogul, U., 2009, “Sentiment Analysis In Turkish.” Middle East Technical University, Ankara

Freud, Y., and Shapire, R. 1996. “Experiments with a New Boosting Algorithm,” in *Machine Learning: Proceedings to the Thirteenth International Conference*, Morgan Kaufmann, San Francisco, CA, pp. 148-156.

Ghag, K, Shah, K. 2013. *“Comparative Analysis of the Techniques for Sentiment Analysis.”* ICATE 2013 Paper Identification Number-124.

Go, A., Bhayani, R., and Huang, L. 2009. “Twitter Sentiment Classification using Distant Supervision,” Stanford University, Palo Alto, CA.

Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*, Springer Science and Business Media, New York, NY.

Hsu, R., See, B., and Wu, A. 2010. “Machine Learning for Sentiment Analysis on the Experience Project,” Stanford University, Palo Alto, CA.

Islam, M.R.. 2014. *“Numeric Rating of Apps on Google Play Store by Sentiment Analysis on User Reviews.”* International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) 2014.

Nigam, K., Lafferty, J., and McCallum, A. 1999. “Using Maximum Entropy for Text Classification,” IJCAI-99 *Workshop on Machine Learning for Information Filtering*, pp. 61–67.

Pang, B., Lee, L.,. and S. Vaithyanathan. 2002. “Thumbs up? Sentiment Classification using Machine Learning Techniques,” *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 79-86.

Razzaq M.A. 2014. *“Prediction and Analysis of Pakistan Election 2013 Based on Sentiment Analysis.”* 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014).

Rosas, VP., 2013. *“Multimodal Sentiment Analysis of Spanish Online Videos.”* IEEE INTELLIGENT SYSTEMS.

Sarlan A, Basri S, Nadam C. 2014. *“Twitter Sentiment Analysis.”* International Conference on Information Technology and Multimedia (ICIMU), November 18 – 20, 2014, Putrajaya, Malaysia.

Setty S,. 2014. *“Classification of Facebook News Feeds and Sentiment Analysis.”* IEEE Computer Society.

Singh, V.K. 2013. *“Sentiment Analysis of Movie Reviews.”* IEEE Computer Society.

Socher, R. 2014. *Recursive Deep Learning for Natural Language Processing and Computer Vision*, “ Dissertation, Stanford University, Palo Alto, CA.

Vapnik, V. 1998. *Statistical Learning Theory*, Wiley, New York, NY.

Venables, W., and Ripley, B. 2002. *Modern Applied Statistics with S, 4th Edition,* Springer, New York, NY.

Wöllmer M., Knaup T, Weninger T, and Morency LP. 2013. *“YouTube Movie Reviews: Sentiment Analysis in an Audio- Visual Context.”* IEEE INTELLIGENT SYSTEMS.

Wang, Z., 2014. *“Anomaly Detection Through Enhanced Sentiment Analysis on Social Media Data.”* 2014 IEEE 6th International Conference on Cloud Computing Technology and Science.

Xia R, Zong C, Xu F, Li T. 2015. *“Dual Sentiment Analysis: Considering Two Sides of One Review.”* Ieee Transactions on Knowledge and Data Engineering.