Twitter Sentiment Classification Using Machine Learning Techniques for Stock Markets

Mohammed Qasem, Ruppa Thulasiram, Parimala Thulasiram
Department of Computer Science
University of Manitoba
Winnipeg, Canada
qasemm@myumanitoba.ca, tulsi@cs.umanitoba.ca, thulasir@cs.umanitoba.ca

Abstract— **Sentiment classification of Twitter data has been successfully applied in finding predictions in a variety of domains. However, using sentiment classification to predict stock market variables is still challenging and ongoing research. The main objective of this study is to compare the overall accuracy of two machine learning techniques (logistic regression and neural network) with respect to providing a positive, negative and neutral sentiment for stock-related tweets. Both classifiers are compared using Bigram term frequency (TF) and Unigram term frequency - inverse document term frequency (TF-IDF) weighting schemes. Classifiers are trained using a dataset that contains 42,000 automatically annotated tweets. The training dataset forms positive, negative and neutral tweets covering four technology-related stocks (Twitter, Google, Facebook, and Tesla) collected using Twitter Search API. Classifiers give the same results in terms of overall accuracy (58%). However, empirical experiments show that using Unigram TF-IDF outperforms TF.**

Keywords—Predictive Modeling;Neural Networks; Logistic Regression; Sentiment; Twitter; Stock Market; Term Frequency; Inverse Document Term Frequency

I. INTRODUCTION AND MOTIVATION

Can Twitter collective sentiment predict the stock market trends? Sentiment mining of Twitter has been shown to be useful in a variety of fields. For instance, it has been successfully applied to predict box office revenues and political poll outcomes [1]. However, applying sentiment analysis of Twitter data for stock-market prediction is still in its infancy and poses more challenges. From the perspective of text classification, these challenges include high degree of language informality, limited length of text, sarcasm and irony, language style and most importantly, what is considered positive news to one person may be considered negative to another. Also, in finance, both random walk and efficient market hypotheses [2, 3] state that stock-market prediction is not possible. The random walk hypothesis argues that stock-market prices change based on a Wiener process, and therefore they are unpredictable [2]. In addition, Efficient-Market Hypothesis (EMH) asserts that all relevant information about a stock are reflected in its current price; thus, stocks are always traded at their fair values. This hypothesis indicates that historical information of stock is irrelevant to its future price; therefore, trying to predict stock prices via technical or fundamental analysis is meaningless [3].

On the contrary, many studies have shown that stock prices are not random, and they ought to be predictable [4, 6, 7]. For instance, one significant study has been recently proposed by Bollen et al. [4]. It involves analyzing the public emotional states, represented by large-scale Twitter feeds to predict trends in Dow Jones Industrial Average (DJIA) over time. Two sentiment analyzers are used, namely Opinion-finder which classifies tweets into positive and negative classes, and Google-Profile of Mood States (GPOMS) which classifies sentiment into six classes (Calm, Alert, Sure, Vital, Kind, and Happy). The study shows that incorporating collective sentiment on Twitter using GPOMS in DJIA prediction can significantly improve prediction accuracy. According to their results, daily changes in closing values of DJIA can be predicted with 86.7% accuracy and Mean Average Percentage Error (MAPE) is reduced by more than 6%.

Twitter can be defined as a real-time microblogging platform that allows people to communicate with short messages. Twitter post, which is known as tweet comprises of 140 characters or less, users can instantly post anything to express their curiosity, thoughts or opinions about any topic – from what they think to how they feel, or just to give status updates to friends and family members. There are many characteristics that make Twitter excellent data source to be considered for sentiment analysis, especially for the purpose of predicting stock market trends. First, unlike other social networking services, Twitter is open for public consumption. Any tweet can be retrieved without any privacy restrictions. Second, twitter has clean and well-documented API that enables developers to query for specific collection of tweets using certain keywords or based over a period of time. Third, Twitter data is particularly interesting because tweets are posted at the "speed of thought" and are available for consumption as they posted in near real time. Finally, Twitter aggregates users into communities and links users in a variety of ways, ranging from short dialogues to interest graphs [5].

978-1-4799-8792-4/15/$31.00 ©2015 IEEE

Logistic regression and Artificial Neural Networks (ANN) models are the most frequently used models for data classification tasks. Unlike other classification methods, such as Support Vector Machines (SVM), decision trees or k-nearest neighbor, These models express the probability of outcome as a liner predictor function $f(x,\theta)$ such that,

$$p(y|x) = f(x,\theta)$$

θ is a vector of parameters which are usually estimated by maximum-likelihood technique for a given dataset. In logistic regression, $f(x,\theta)$ is known as a parametric method, whereas in neural networks is called semi-parametric or non-parametric [18]. Generally, logistic regression predicts the probability of occurrence of an event by fitting data to a logistic function. For instance, if outcome variable y is binary (0, 1), logistic function is given by:

$$p(1|x,\theta) = \frac{1}{1+e^{-z}}$$

and $p(0|x,\theta) = 1 - p(1|x,\theta)$ here, z is a linear function of the predictor variables, such that: $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$ where β_0 is constant, $\beta_{1,2,...,p}$ are predictor variable coefficients or regression coefficients and $x_{1,2,...,p}$ are predictor variable values. This logistic transformation forces probability estimates to be between 0 and 1 regardless of the value of z. The predictor variable coefficients are computed using a maximum-likelihood technique. . If the estimated probability of the event under consideration is less than 0.5, then it is concluded that the event will not occur. In contrast, if the estimated probability of the event is greater than 0.5, it is inferred that the event will occur. If, however, the estimated probability exactly equals 0.5 then no inference concerning the occurrence of the event can be made.

Multiclass logistic regression is an extension of binary logistic regression. There are many features to use multiclass logistic regression for text classification. First, it is applicable when outcome variable is to be classified to one of multiple possible categories. Second, unlike liner regression model or general linear regression model, logistic regression can be used whether independent variables are statistically independent or not. Third, independent variables can be discrete or continuous. Forth, it can handle non-linear relationships between dependent variable and independent variables. Lastly, no assumptions regarding linearity, normality or homoscedasticity are required.

ANNs are a family of non-linear statistical learning algorithms inspired by biological neural networks. It consists of a number of elementary processing units (known as neurons). These neurons are interconnected and arranged in layers. Multilayer NN may include one or more hidden layers. Data flow through the network during the learning phase from input layer to the output layer to produce the solution. Each neuron produces output by adding weights from one or more inputs. These weights determine the strength of each input connected to a given neuron. Generally, NN is trained by back propagation technique where inputs and outputs of the problem under consideration are fed to the input layer. The neurons process the input data, and then propagate the values resulting from each neuron through the network. Once the values reach the output layer, the output which is computed by the network, is compared with the desired output. Any error is used for adjusting connection weights, working backwards through the network. The weights which connect the neurons are adjusted by promoting connections which produce correct answers and demoting those which produce incorrect answers. This process continues until the network learned the relationship between inputs and desired outputs. Once weights are stabilized, the resulting network can be employed for the designated purpose.

In this work, we model tweet collection using vector space model (SVM). In this model, each tweet is represented as a vector of weights in the term-space. We calculate term weights using term frequency (TF) and term frequency –inverse document term frequency (TF-IDF). TF measures how frequently a term (feature) occurs in a tweet. Since every tweet may have different length, it is possible that a term would appear much more times in long tweets than shorter ones. Thus, the term frequency is often divided by the tweet length (the total number of terms in the tweet) as a way of normalization. So, normalized TF for a given term t is defined as:

TF(t) = (Number of times term t appears in a tweet) / (Total number of terms in the tweet).

In contrast, IDF measures the importance of terms based on how frequently they appear across multiple tweets. Intuitively, a term that appears frequently in a tweet is important and gets a high weight. However, if the term appears in many tweets, then it becomes less discriminative; hence, IDF deemphasizes its weight. IDF for a term t is given by:

IDF(t) = Log (Total number of tweets / Number of tweets with term t in it).

TF-IDF for a certain term t is defined as the multiplication of TF(t) by IDF(t).

This work contributes to the sentiment classification by addressing the accuracy of using multiclass neural network and multiclass logistic regression in classifying financial tweets into positive, negative and neutral classes. It also compares the accuracy of these models using two feature space models: Bigram TF and Unigram TF-IDF.

II. Background and Related Work

The characteristics of Microblog data like Twitter, poses different challenges on sentiment classification, from the limited length of text to the high degree of language informality. Previous works of sentiment analysis fall into two main directions. The first one is directed to the problem of sentiment classification of general Twitter data using machine learning techniques, whereas the second direction concentrate on employing sentiment classification of financial tweets for the purpose of stock price prediction.

A. Sentiment classification

In terms of sentiment classification of general tweets, Go et al. [10] propose the idea of using emoticons for sentiment classification of Twitter data. They construct training data automatically by using positive emoticons like ":)" to classify positive tweets, and negative emoticons like ":(" to classify negative tweets. They experiment different machine learning algorithms using distant supervised learning, such as Naïve Bays, Maximum Entropy and SVM. Experiments include Unigram, Bigram Features with part-of-speech tagging. According to their results, accuracy reaches more than 80%.

Pak and Paroubek [12] follow the same procedure of Go et al. [10] to collect training data. However, they enhance training data by identifying objective tweets, which are considered to bear neutral sentiment. They collect these tweets using search queries from popular newspapers and magazines, such as "New York Times" and "Washington Posts". They apply sentiment classification using the multiclass Naïve Bays. Features are represented using N-gram with Part-of-Speech tagging. Their results show that highest accuracy is achieved by using POS and Bigrams.

Bermingham and Smeaton, [11] compare the accuracy of Support Vector Machine (SVM) versus Multiclass Naive Bayes (MNB) in classifying sentiment of microblogs. Training data are gathered for five categories: Entertainment, Products and Services, Sport, Current Affairs and Companies. Training data are annotated manually. According to their results, SVM accuracy for microblogs reaches 74.85%.

Another significant study for sentiment classification on Twitter data is proposed by Barbosa and Feng [13]. They use features that are extracted from tweet text, such as hashtags, link, punctuation and exclamation marks along with word features that bear polarity. Classifiers are tested and trained using 2000 manually annotated tweets. However, they do not explain how they gather their test data. Barbosa et al. [14] extend their approach by using real valued prior polarity by combined with POS. Their results show that performance of classifiers can be significantly enhanced by combining prior polarity of words with their parts-of-speech.

B. Sentiment Classification for Stock Price Prediction

Most recent studies in this direction include [15, 16, 17]. They focus on using sentiment classification of Twitter data to predict stock price. For example, Schumaker and Chen [15] perform text classification on articles of financial news to forecast the price of S&P500 stocks. They construct a corpus that contains 9,211 articles of financial news and 10,259,042 stock quotes covering the S&P 500 stocks over a five week period. The estimation of stock price is performed twenty minutes after a news article is released. Features are represented using vector space model, noun phrases and named entities. They experiment using different models including linear regression model and another three models that use support vector machine (SVM). According to their results, the best performance is achieved when article terms are correlated with the stock price at the release of the article.

Tayal and Komaragiri [16] explore and compare traditional blogs with microblogs to determine which one has more predicative power on stocks. Their main goal is to determine which of these formats is more useful for autonomous stock price predictor. They apply sentiment classification of blogs and microblogs. According to their experiments, predictive accuracy of microblogs outperforms blogs. The used data sources are obtained from the web service Google Blogsearch11 and Twitter. Sentiment analysis are carried out using a lexicon of positive and negative terms. They predict the actual stock price of the following day from using ach data source. The results also show that Twitter data source gives more concise sentiment results due to the fact that each tweet is limited to 140 characters, and it encapsulates a single topic.

Smailovic et al. [17] propose an active learning approach for sentiment analysis of tweet streams in the stock market domain. The proposed methodology is designed for stream-based active learning of tweet sentiment analysis in finance. The methodology continuously changing tweet streams to determine the best querying strategy for active learning of the SVM classifier, which is adapted to sentiment analysis of streams of financial tweets and applied to predictive stream mining in a financial stock market application. In addition, they have labeled and made publicly available collection of financial tweets since there is no large labeled dataset of financial tweets publicly available. This labeled dataset is used in the simulated active learning setting and in the evaluation of the results of tweet stream analysis.

In the next section, we present a brief

C. Cloud Computing and Microsoft Azure

Cloud computing has become popular computing model in business and academia. It aims to provision end users with computing services over the Internet. These computing services are provisioned as on-demand self-services based on pay-as-you-go manner. They are characterized with high scalability, rapid elasticity and guaranteed quality of service. Different definitions of cloud computing have been proposed based on different perspectives. For instance, business definitions of cloud computing include [18, 19, 20]. These definitions define cloud computing from the perspective of the end users. Their focus is on how cloud computing might be experienced by them, and they consider the core feature of Cloud computing as the provision of IT Infrastructure and applications as scalable services. On the hand, scientific community defines cloud computing [21, 22] from the perspective of service providers by considering datacenters as the major component of the cloud, or from the perspective of cloud purpose by classifying clouds onto public, private, community and hybrid. However, a wildly cited definition proposed by the U.S. National Institute of Standards and Technology (NIST) September, 2011 [22]. It states that:

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Generally, cloud computing are divided into two distinct sets of models. The first set includes cloud models based on their deployment. Cloud deployment refers to the location and management of the cloud's infrastructure. These models include public clouds where infrastructure are made available for public use. Private clouds where infrastructure is owned and used exclusively by an organization. Both public and private clouds can be combined to form Hybrid clouds. If the cloud is organized to serve a common function or purpose, then it is called community cloud [23]. The second set includes service oriented models or what is known as XaaS "<Something> as a Service". These services, which are provided clouds in this set of models are categorized into three major classes: software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service. (IaaS) provides clients with infrastructure resources based on their demand. These resources are provided in the form of virtual machines. A client of IaaS is provided with virtual computing resources, storage space, or network resources to run platform systems, or applications. In PaaS, clients are provided with hardware and other platform-layer resources, such as operating system support, development frameworks software development toolkits and selected programming languages. Examples of such type include Microsoft Azure and Google AppEngine. In the SaaS model, also known as the Application Service Provider model, Clients are considered end-users of applications running on a cloud infrastructure and may only control application parameters for specific user settings [8][10]. The applications are typically accessed through thin-client interfaces, such as web browsers [23].

In February 2001, Microsoft released Windows Azure as a cloud computing platform and infrastructure. Azure supports both PaaS and IaaS services. It also allows programmers to use different programming languages. For example, websites can be developed using ASP.NET, PHP, Python, or Node.js. In addition, Azure provides Customers with Windows Server and Linux Virtual Machines. Microsoft's Platform as a Service (PaaS) provides customers with a wide variety of services and can be used to create scalable applications and services. Applications can be public web applications (such as web sites and e-commerce solutions), or they can be dedicated for specific tasks like processing orders or analyzing data. Data management are implemented by Microsoft SQL Server technology. It also integrates with Active Directory, Microsoft System Center and Hdinsight [25].

We create our experiments using Azure Machine Learning (Azure ML) which is cloud service designed for predictive analytics. It provides capabilities of new analytics tools, powerful algorithms developed for Microsoft products like Xbox and Bing, and years of machine learning experience into one simple and easy-to-use cloud service. Predictive models on Azure ML studio can be deployed interactively and easily. Modeling steps include data preparing, creating model using various machine learning algorithms, evaluating models based on their accuracy to predict correct outputs, deploying and finally testing and using. Azure ML provides clients with black-box modules for each development phase. In addition, it enables model customization by providing Python and R-script Modules. Also, Azure ML provides many machine learning algorithms including classification, regression and clustering. Finally, Azure ML experiments are deployed as cloud services so they can be used in websites, desktop applications or in other cloud services [25].

B. Objectives

Sentiment analysis modeling has become easier to accomplish by utlizing cloud computing services. For instance, Azure ML enables the deployment of different cloud services to to retrive hetrogenous data for different sources. It also dedicates services for text preprocessing and entity extraction. Text can be classified via distict classification and clustering algorithms so that sentiment models can be scored, evaluated and tested.

The main objective of this work is to measure overall accuracy of sentiment classification of financial-tweet datasets using neutral network and logistic regression classifiers. Both models are evaluated under two feature spaces: Bigram TF and Unigram TF-IDF using Microsoft Azure ML.

III. FEATURE SELECTION

Sentiment analysis refers to the process of finding and analyzing personal information. Personal information include opinions, evaluations and attitudes. They also involve sentiments, appraisals as well as emotions [1]. In this work, we are interested in finding sentiments of financial tweets by classifying these tweets into positive, negative or neutral classes. A financial tweet can be defined as a tweet in which its text contains one or more cashtags. According to twitter language style, cashtags are identified by "$" followed by stock symbol or ticker symbol. For example, a financial tweet "Stock Market Analysis Video with Mike Cintolo (@cabotdude): http://bit.ly/dGUUaC $FNSR $CRM $FFIV $SINA $OPEN $BIDU $LULU $RL $PCLN $UA" contains 10 cashtags. In addition to the cashtags, tweet text may include the following parts:

- Usernames: the convention is the use of @twitter_user to mention other user in tweet text.
- Web links: short URLs style is used to link a website or a webpage
- Hashtags "#":are used for specifying tweet topic
- Cashtages "$": are used for indicating symbol of a stock e.g. $goog
- Letter repetitions: such as greaaaaat instead of great
- Negations: like wasn't, aren't, hasn't…..etc.
- Exclamation and question marks: examples like "???" or "!!!"
- Emoticons: like ":)", "=)" or ":("

A major challenge for text classification is feature selection, whereby an attempt is made to determine the most relevant features to the classification process. This challenge is attributed to the fact that certain tweet parts are expected to be more correlated to the class distribution than others. Since the tweet parts represent a potential features, we need to measure the extent to which we can depend upon these features to train the classifiers (neural network or logistic regression) so that they can classify new instances.

IV. PEROFRMANCE STUDY WITH MS. AZURE ML

The major task in this project is to compare the accuracy of using neural network and logistic regression classifiers to find sentiment polarity of large number of financial tweets retrieved by Twitter Search API. Since there is no manually annotated training datasets publically available, a training dataset has been constructed using the technique proposed in the technical report written by [10], the automatic annotation process works as follows:

- if tweet contains any one of positive emoticon :),:-), :D, =), :) or =D then, it bears positive sentiment
- if tweet contains negative emoticons :<, :-(, : (or :(then, it bears negative polarity
- Tweets that do not contain emoticons or keywords that bear polarity were assumed to have neutral sentiment
- Tweets containing both positive and negative emotions were removed

By querying Twitter using the above rules for each stock symbol, we constructed a training dataset of 42,000 tweets equally divided into the three predefined classes. Each instance in dataset consists of the following fields: Tweet ID, Created at (UTC time format), Creator, Tweet Text, Favorites, Hashtags, User_ Mentions, URLs, Polarity. An example of that "583615136365019000, 02/04/2015 7:01, $FB Omg on this $C upgrade. BREAKING NEWS! Clarity upgrades Facebook to a buy target $86.07. XD :) :) Joshua Sanchez, 0, 0, 0, 0, 0, P". All emoticons are filtered so that classifiers can learn from other possible features in tweet text, such as user mentions, URLs, hashtags…etc. we experiment with two different feature extractors. The first one is a Unigram TF-IDF, and the second is Bigram TF. The following complete data processing pipeline illustrates the entire experimental framework (Fig 1) on Windows Azure Machine Learning Studio (Azure ML):

Step 1: Data Preparation

Dataset is cleaned by removing entire row if polarity is empty i.e. tweet text dos not contain emoticons to determine its polarity.

Step 2: Text Preprocessing

In this step, special characters and duplicate letters in text are replaced, stop words are filtered, and remaining terms are stemmed.

Step 3: Feature Engineering

Two feature extractors are implemented in this step. The first one is a Bigram TF, and the second one is a Unigram TF-IDF. In both cases, features are ranked using Chi-squared, then we filter 5000 top features for dimensionality reduction purposes.

Step 4: Training and Evaluate Models

In this step, we train multi-class logistic regression and neural network classification models using 70% of dataset while the remaining 30% is divided equally for validation set and test set. Azure ML provides Sweep Parameters module to get the optimal values for the underlying learning algorithm parameters. The parameter sweeping mode is set to "entire grid" where the module will conduct a number of training runs to explore all possible values for each parameter of the underlying classification modules.

V. RESULTS

A. Performance Measures for Multiclass Classification

The results for each multiclass text classifier under consideration are provided using the measures listed in table 1. For each class C_i, the assessment is measured by the number of correct examples identified as a part of the class (true positives), the number of correct examples that were not recognized as a part of the class (true negatives), and examples that either were incorrectly assigned to the class (false positives) or that not recognized as class examples (false negatives) [17]. These count percentages are listed in the form of confusion matrices (Table 2, 3)

Precision is the number of correctly classified examples divided by the number of examples labeled by the system as positives; whereas, recall is the number of correctly classified positive examples divided by the number of positive examples in the dataset [17]. For each individual class C_i, the assessment is defined by $tp_i, fn_i, tn_i, fp_i, accuracy_i, Precision_i, recall_i$ which are calculated from the counts of each class C_i. The overall quality of classifier is measured in two ways: Macro-averaging which is the average of measures calculated for each class C_i , or Micro-averaging which is the sum of counts to get cumulative tp, fn, tn, fp and then calculating performance measure [17].

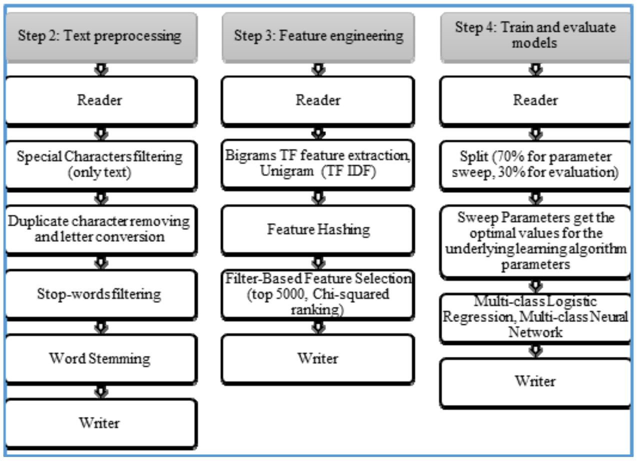
Step 2: Text preprocessing
Reader
Special Characters filtering (only text)
Duplicate character removing and letter conversion
Stop-words filtering
Word Stemming
Writer
Step 3: Feature engineering
Reader
Bigrams TF feature extraction, Unigram (TF IDF)
Feature Hashing
Filter-Based Feature Selection (top 5000, Chi-squared ranking)
Writer
Step 4: Train and evaluate models
Reader
Split (70% for parameter sweep, 30% for evaluation)
Sweep Parameters get the optimal values for the underlying learning algorithm parameters
Multi-class Logistic Regression, Multi-class Neural Network
Writer

Fig1: Experimental Framework

TABLE 1: Measures for multi-class classification, counts for each class C_i tp_i: true positive, fp_i: false positive, fn_i: false negative, tn_i: true negative respectively N: total number of instances , l: number of classes [17]

Measure	Formula	Evaluation Focus
Overall Accuracy	$\frac{\sum_{i=1}^{l} tp_i}{N}$	overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications
Average Accuracy	$\frac{\left(\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i}\right)}{l}$	The average per-class effectiveness of a classifier

Precision (Micro)	$\frac{(\sum_{i=1}^{l} tp_i)}{(\sum_{i=1}^{l} tp_i + fp_i)}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions
Recall (Micro)	$\frac{(\sum_{i=1}^{l} tp_i)}{(\sum_{i=1}^{l} tp_i + fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions
Precision (Macro)	$\frac{(\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i})}{l}$	An average per-class agreement of the data class labels with those of a classifiers
Recall (Macro)	$\frac{(\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i})}{l}$	An average per-class effectiveness of a classifier to identify class labels

B. *Performance Measures under TF and TF-IDF*

Fig 2 and Fig 3 show the results of neural network and logistic regression using TF and TF-IDF weighting schemes respectively. In both models, TF-IDF scores higher overall accuracy than TF. This indicates that financial tweets of a certain stock almost contain the same features (URLs, User Mentions, hashtags and cachtags); therefore, they can be better distinguished by reducing the effect of repeated features. Typically, the TF-IDF weight constitutes two factors: the first is the normalized Term Frequency (TF), which is the number of times a term appears in a tweet text, divided by the total number of terms in that tweet; the second factor is the Inverse Document Frequency (IDF), which is computed as the logarithm of the total number of the tweets in the corpus divided by the number of tweets where the specific term appears.

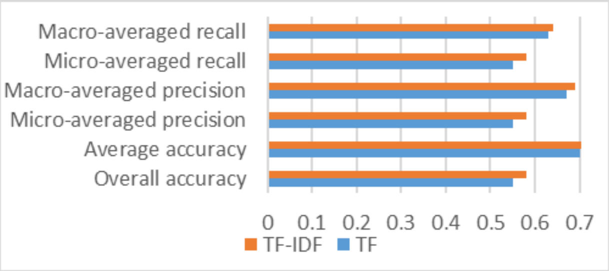

Fig 2: Performance measures of neural network for TF and TF-IDF

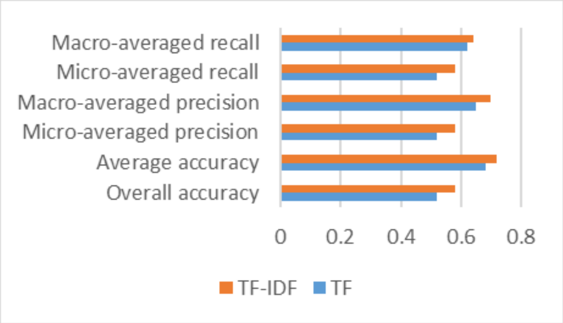

Fig 3: Performance measures of logistic regression for TF and TF-IDF

TABLE 2: Performance measures Logistic Regression vs. Neural Network

	Neural Network		Logistic Regression	
Metric	**TF**	**TF-IDF**	**TF**	**TF-IDF**
Overall accuracy	0.52	0.58	0.55	0.58
Average accuracy	0.68	0.72	0.70	0.72
Micro-averaged precision	0.52	0.58	0.55	0.58
Macro-averaged precision	0.65	0.70	0.67	0.69
Micro-averaged recall	0.52	0.58	0.55	0.58
Macro-averaged recall	0.62	0.64	0.63	0.64

C. *Confusion Matrices*

Tables 3 and 4 show the results of confusion matrices using TF, TF-IDF weighting schemes. It can be noticed that neural network outperforms logistic regression in predicting Negative Class instances (83%) using TF weighting scheme. However, logistic regression achieves (82.3%) under TF-IDF. Also, it predicts Positive class instances with (66.1%) correctness. Whereas, neural network achieves only (50.1%) under TF-IDF. Neutral Class represents weakness point in both models (68%: neural network and 52.6% in logistic regression). Nevertheless, neural network scores the maximum correctness under TF-IDF. However, this weakness in predicting neutral tweets is due to the high conflict with positive class. This means that neutral are poorly identified in training set.

VI. CONCLUSION

In this work, two machine learning techniques are evaluated to classify tweets that are related to four major stocks (TWTR, GOOG, FB and TSLA) into positive, negative and neutral classes. Both models are tested under two weighting schemes; Unigram term frequency (TF) and Bigram term frequency-inverse term frequency (TF-IDF).Training dataset for both models is collected using Twitter Search API. It contains 42,000 tweets equally divided into the predefined three classes. Positive tweets are defined as those tweets which contain positive emoticons. Similarly, negative tweets contain negative emoticons, and neutral tweets do not have emoticons or keywords that indicate polarity, such as happy, sad, good, rise, down ...etc. According to the results, Unigram TF-IDF weighting outperforms Bigram TF in both models in terms of overall accuracy (58%). However, a major weakness

TABLE 3: Confusion matrix: Multiclass neural network

		Predicted Classes					
		Negative		**Positive**		**Neutral**	
		TF	**TF-IDF**	**TF**	**TF-IDF**	**TF**	**TF-IDF**
Actual Classes	**Negative**	83%	76%	14.7%	21.7%	2.3%	2.3%
	Positive	0.7%	0.3%	37.3%	50.1%	62%	49.6%
	Neutral		0.1%	31.8%	34.6%	68.2%	65.3%

TABLE 4: Confusion matrix: Multiclass logistic regression

		Predicted Classes					
		Negative		**Positive**		**Neutral**	
		TF	**TF-IDF**	**TF**	**TF-IDF**	**TF**	**TF-IDF**
Actual Classes	**Negative**	81%	82.3%	17%	16.3%	2%	1.3%
	Positive	0.3%	0.4%	55%	66.1%	44.6%	33.5%
	Neutral			47.4%	55.2%	52.6%	44.8%

point in both classifiers is the high conflict between positive and neutral classes. This conflict is due to the automatic annotation of classes, especially the neutral class. As a future work, we expect that clustering techniques could enhance the results since it do not require a training dataset.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval,* vol. 2, pp. 8-10, 2008.

[2] B. G. Malkiel, *A Random Walk Down Wall Street: Including a Life-Cycle Guide to Personal Investing.* WW Norton & Company, 1999.

[3] E. F. Fama, "The behavior of stock-market prices," *Journal of Business,* pp. 34-105, 1965.

[4] J. Bollen, H. Mao and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science,* vol. 2, pp. 1-8, 2011.

[5] M. A. Russell, *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More.* O'Reilly Media, Inc, 2013.

[6] L. Bing, K. C. Chan and C. Ou, "Public sentiment analysis in twitter data for prediction of a company's stock price movements," in *E-Business Engineering (ICEBE), 2014 IEEE 11th International Conference on,* 2014, pp. 232-239.

[7] M. Mittermayer, "Forecasting intraday stock price trends with text mining techniques," in *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on,* 2004, pp. 10-pp.

[8] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP),* 2013, pp. 1642.

[9] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management,* vol. 45, pp. 427-437, 2009.

[10] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, Stanford, pp. 1-12, 2009.

[11] A. Bermingham and A. F. Smeaton, "Classifying sentiment in microblogs: Is brevity an advantage?" in Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 1833-1836.

[12] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *Lrec,* 2010, pp. 1320-1326.

[13] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters,* 2010, pp. 36-44.

[14] C. Oh and O. Sheng, "Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement," 2011.

[15] D. Tayal and S. Komaragiri, "Comparative Analysis of the Impact of Blogging and Micro-blogging on Market Performance," *International Journal,* vol. 1, pp. 176-182, 2009.

[16] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management,* vol. 45, pp. 427-437, 2009.

[17] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *J. Biomed. Inform.,* vol. 35, pp. 352-359, 2002.

[18] Gartner, http://www.gartner.com/newsroom/id/766215 accessed on Mar 30, 2015.

[19] IDC, http://blogs.idc.com/ie/?p=190, accessed on Mar 30,2 015.

[20] The 451 Group, www.451group.com/reports/execu=ve_summary.php?id=619, accessed on Mar 30, 2015.

[21] Armbrust, Michael and Fox, Armando and Griffith, Rean and Joseph, Anthony D and Katz, Randy and Konwinski, Andy and Lee, Gunho and Patterson, David and Rabkin, Ariel and Stoica, Ion and others, "A view of cloud computing," *Commun ACM,* vol. 53, pp. 50-58, 2010.

[22] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.

[23] B. Sosinsky, *Cloud Computing Bible.* John Wiley & Sons, 2010.

[24] Microsoft windows Azure, http://www.windowsazure.com/en-us/ accessed on Mar 30, 2015.