

RESEARCH

Open Access



# Big Data: Deep Learning for financial sentiment analysis

Sahar Sohangir<sup>1\*</sup> , Dingding Wang<sup>1</sup>, Anna Pomeranets<sup>2</sup> and Taghi M. Khoshgoftaar<sup>1</sup>

\*Correspondence:

ssohangir2014@fau.edu

<sup>1</sup> Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA  
Full list of author information is available at the end of the article

## Abstract

Deep Learning and Big Data analytics are two focal points of data science. Deep Learning models have achieved remarkable results in speech recognition and computer vision in recent years. Big Data is important for organizations that need to collect a huge amount of data like a social network and one of the greatest assets to use Deep Learning is analyzing a massive amount of data (Big Data). This advantage makes Deep Learning as a valuable tool for Big Data. Deep Learning can be used to extract incredible information that buried in a Big Data. The modern stock market is an example of these social networks. They are a popular place to increase wealth and generate income, but the fundamental problem of when to buy or sell shares, or which stocks to buy has not been solved. It is very common among investors to have professional financial advisors, but what is the best resource to support the decisions these people make? Investment banks such as Goldman Sachs, Lehman Brothers, and Salomon Brothers dominated the world of financial advice for more than a decade. However, via the popularity of the Internet and financial social networks such as StockTwits and SeekingAlpha, investors around the world have new opportunity to gather and share their experiences. Individual experts can predict the movement of the stock market in financial social networks with the reasonable accuracy, but what is the sentiment of a mass group of these expert authors towards various stocks? In this paper, we seek to determine if Deep Learning models can be adapted to improve the performance of sentiment analysis for StockTwits. We applied several neural network models such as long short-term memory, doc2vec, and convolutional neural networks, to stock market opinions posted in StockTwits. Our results show that Deep Learning model can be used effectively for financial sentiment analysis and a convolutional neural network is the best model to predict sentiment of authors in StockTwits dataset.

**Keywords:** Deep Learning, Big Data, Sentiment analysis, Information retrieval

## Introduction

The Internet, as a global system of interconnection, provides a link between billions of devices and people around the world. The rapid development of social networks causes the tremendous growth of users and digital content [1]. It opens opportunities for people with various skills and knowledge to share their experiences and wisdom with each other. There are many websites like *Yelp*, *Wikipedia*, *Flickr*, etc. that use the power of the Internet to help their users make optimal decisions.

Furthermore, there are websites that give users the ability to consult with professionals, and one topic that is always popular is investment. Companies like Goldman Sachs

and Lehman Brothers have more than 150 years of investment advice. In the Internet age, independent analysts and retail investors around the world can collaborate with each other through the web. Seeking Alpha and StockTwits are two examples of common financial social media platforms focused on the stock market, giving their users a way to connect with information and each other and grow their investments [2].

Financial social media brings people, companies, and organizations together so that they can generate ideas and share information with others. It is this media that provides a huge amount of unstructured data (Big Data) that can be integrated into the decision-making process. Such a Big Data can be considered as a great source of real-time estimation because of its high frequency of creation and low-cost acquisition.

*Sentiment analysis (SA)* is a common method which is increasingly used to assess the feelings of social media users towards a subject. The most popular approach performing sentiment analysis is using data mining. Our central idea is to adopt Deep Learning to determine investors' expectations about the price of stocks and the overall market based on their messages. The reason why we select Deep Learning methodology rather than data mining is that in data mining, identifying features and selecting the best of those features is the most challenging task to undertake especially in a Big Data.

In contrast to data mining, a Deep Learning model, learns features during the process of learning. Deep Learning algorithms lead to abstract representation, as a result, they can be invariant to the local change in the input data. In addition, Big Data problems including semantic indexing, data tagging, and fast information retrieval can be addressed better with the aid of Deep Learning. Deep Learning provides the opportunity to use a simpler model to accomplish complicated Artificial Intelligence tasks. Although Deep Learning algorithms have been used for some Big Data domain like computer vision [3–5, 17, 18, 22] and speech recognition [6–10, 11] it is still intact in the context of Big Data analysis. In this paper, we evaluate the adoption of Deep Learning for sentiment analysis of financial data.

Deep Learning algorithms provide the opportunity to extract complex data at a high level of abstraction in a way that high-level features with more abstraction are defined in terms of lower-level features with less abstraction [12]. A different source of variation in data (like light, object shapes, and object materials in an image) can be separated by using Deep Learning. The idea of hierarchical learning in Deep Learning coming from the primary sensorial areas of the neocortex in the human brain [13].

Convolutional neural network (CNN) [14] is an example of a various number of Deep Learning models. This model which is popularly used for image analysis can make use of the internal structure of data through convolution layers. Because of the internal structure that exists inside the text documents CNN has been gaining attention on text data as well. CNN is used in systems for tagging, entity search, sentence modeling, etc. [15–23].

Deep Learning algorithms which usually learn data representations in a greedy fashion, look more useful to learn from Big Data [24, 25]. Deep Learning can be used to extract nonlinear complicated features in Big Data analytics, then extracted features are used as input to a linear model.

Deep Learning can be used to make discriminative tasks of Big Data analytics easier. For instance Li et al. [26] use Deep Learning to do a search in Big Data. They use Deep

Learning to enable searching of audio and video file with speech. Deep Learning ability to extract high-level, complex abstractions from large volumes of unsupervised data (Big Data) make it desirable for Big Data analytics. High-level features can be extracted from unlabeled images by using Deep Learning. For instance, Google provides a deep neural network that can learn high-level features from unlabeled data [27, 28]. Their work clearly shows how Deep Learning methods can extract high-level features from unsupervised data and demonstrates the advantages of using Deep Learning with unsupervised data (Big Data).

The remainder of this paper is organized as follows: “[Related work](#)” section we look at previous work on financial sentiment analysis and the methods employed therein; “[Big Data](#)” section contains an overview of Big Data analytics. In this section, we discuss some Big Data characteristics and specify main problems that faced Big Data in data analysis; “[Sentiment analysis](#)” section we briefly talk about sentiment analysis methods and advantages of using Deep Learning in sentiment analysis; in “[Sentiment analysis with data mining approaches](#)” section we explore other works that use data mining to do sentiment analysis on StockTwits dataset. After that, we discuss some feature selection methods. In “[Deep Learning in Big Data analytics](#)” section we explore how Deep Learning can be used for Big Data analysis also we discuss some challenges that Deep Learning needs to overcome to do analysis in the Big Data domain; “[Results and discussion](#)” section explains our experiments, and goes into depth about how we can apply Deep Learning to financial sentiment analysis. Our primary findings and conclusions are presented in “[Conclusions](#)” section.

## Related work

Specific Big Data domains including computer vision [29] and speech recognition [30], have seen the advantages of using Deep Learning to improve classification modeling results but, there are a few works on Deep Learning architecture for sentiment analysis. In 2006 Alexandrescu et al. [31] present a model where each word is represented as a vector of features. A single embedding matrix is used to look up all of these features. Luong et al. [32] use a recursive neural network (RNN) to model the morphological structures of words and learn morphologically-aware embeddings. In 2013 Lazaridou et al. [33] try to learn meanings of a phrase by using compositional distributional semantic models. In 2013 Chrupala use a simple recurrent network (SRN) to learn continuous vector representations for sequences of characters. They use their model to solve a character level text segmentation and labeling task. A meaningful search space via Deep Learning can be constructed by using Recurrent Neural Network [34] Socher et al. in 2011 [35], use recursive autoencoders [36–39] for predicting sentiment distribution and proposed a semi-supervised approach model. In 2012 Socher et al. [40] propose a model for semantic compositionality with the ability to learn compositional vector representation for sentences of arbitrary length. Their proposed model is a matrix-vector recursive neural network model. Recursive Neural Tensor Network (RNTN) architecture proposed in [41]. RNTN use word vector and a parse tree to represent a phrase and then use a tensor-based composition function to compute vectors for higher nodes [42]

Regarding convolutional network for NLP tasks, Collobert et al. [15] for semantic role labeling task avoid excessive feature engineering by using the convolutional neural

network. In 2011 Collobert used a similar network architecture for syntactic parsing. In [43] a deep convolutional neural network is proposed that exploits the character-to-sentence-level information to perform sentiment analysis of short texts.

The experiments in this paper focus on market sentiment. Based on the definition in [44], market sentiment is the general prevailing attitude of investors as to anticipate price development in a market. This attitude is the combination of various factors such as world events, history, economic reports, seasonal factors, and many others. Market sentiment is found through sentiment analysis, also known as opinion mining [45], which is the use of natural language processing methods to extract the attitude of a writer from source materials.

Wang and Sambasivan in [2] apply market sentiment on the StockTwits dataset by using supervised sentiment analysis classified messages in StockTwits as “Bullish” or “Bearish”. An investor is considered Bullish if he or she believes that the stock price will increase over time and recommends purchasing shares. Oppositely, if an investor is Bearish he or she expects downward price movement and will recommend selling shares or against buying.

One of the most popular works in this field is by Loughran and McDonald [46]. They used the US Security and Exchange Commission portal from 1994 to 2008 to make a financial lexicon and manually create six-word lists including *positive*, *negative*, *litigious*, *uncertainty*, *model strong* and *model weak*. Mao et al. [47] propose an automatic Chinese financial lexicon constructor. His proposed procedure explores many corpora classified as positive or negative and attempts to construct a Chinese financial lexicon automatically.

Supervised classification methods, such as Support Vector Machines [48], Naïve Bayes [49] or ensembles [50, 51] have been deployed to perform sentiment analysis in multiple research projects. Machine learning techniques mainly use the bag-of-words [52] model. In the bag-of-words model, a text is represented as the collection of its words, disregarding the order of those words in their sentences. However, the order of the words in a sentence can change the sentiment of a word. For example, consider the word “underestimate”. This word potentially has a negative connotation, but if we consider it beside other words like “underestimated stock” it can become positive.

Recently, Deep Learning approaches have emerged as a powerful tool in sentiment analysis in Big Data due to the advantages they provide over other methods. One of these advantages is that features are learned hierarchically during the process of Deep Learning instead of the feature engineering that is required in data mining. Additionally, in Deep Learning methods, each word is considered as part of a sentence. In this way, relevant information contained in word order, proximity, and relationships is not lost. Furthermore, Deep Learning benefits from a similarity model. Word embedding creates a vector representation of words with a much lower dimensional space compared to the bag of the words model [53, 54]. The vectors representing similar words in vector space are therefore closer together. One of the other main concepts in Deep Learning algorithm is the automatic extraction of representation (abstractions) [55]. To achieve this goal Deep Learning uses a massive amount of unsupervised data (Big Data) and extracts complex representation automatically. One of the advantages of abstract representation extracted with Deep Learning algorithms is their generalization. Features extracted

from a given dataset can be used successfully for a discriminative task on another dataset. Deep Learning is an important aspect of artificial intelligence because it provides a complex representation of Big Data and also makes the machine independent of human knowledge.

Deep Learning constructs complicated representations for image and video data with a high level of abstraction. High-level data representations provided by Deep Learning can be used for simpler linear models for Big Data. This representation can be useful for image indexing and retrieval. In other words, Deep Learning can be used in the discriminative task of semantic tagging in the context of Big Data analysis.

## Methodology

### Dataset

We were fortunate to receive permission from StockTwits Inc. to have access to their datasets. StockTwits is a financial social network which was established in 2009. Information about the stock market, like the latest stock prices, price movement, stock exchange history, buying or selling recommendations, and so on, are available to StockTwits users. In addition, as a social network, it provides the opportunity for sharing experience among traders in the stock market. Through the StockTwits website, investors, analysts, and others interested in the market can contribute a short message limited to 140 characters about the stock market. This message will be posted to a public stream visible to all site visitors. Moreover, messages can be labeled Bullish or Bearish by the authors to specify their sentiment about various stocks.

In our experiment, we used messages which were posted in the first six months of 2015. Each message includes a messageID, a userID, the author's number of followers, a timestamp, the current price of the stock, and other record-keeping attributes. We examined the posts to see if there is any relation between the future stock price and users' sentiment. In other words, we want to see if we can predict a future stock price based on the current sentiment of many users.

We can use Pearson Correlation Coefficient [6] to see if there is a linear relation between a stock's future price and the user's sentiment. Pearson Correlation is one of the most widely-used functions to measure the linear correlation between two variables. It returns one if there is a perfect positive correlation between the two input variables,  $-1$  if there is a perfect negative correlation, and 0 if there is no correlation. The Pearson Correlation Coefficient between a stock price and a general user's sentiment is equal to 0.05, which means that only 53% of the time are users able to predict future stock prices correctly. This is a little bit better than a random guess, so we will examine whether that accuracy improves if the number of predictions is increased.

Wang in [2] tried to find if there are authors in financial social media whose contributions provide good predictors of stock price, but buried in the noise. They ranked authors based on their performance in predicting stock price within the week of their prediction. They use two consecutive years of data, the first year as a benchmark to find such top authors, and the second year to examine the top authors performance. Based on the results published in [2], the correlation score for top authors is around 0.4, which means that top authors can predict stock price movement with the accuracy of about 75%.

Knowing the sentiment of top authors, we can predict stock prices with accuracy of 75% but unfortunately, only 10% of messages in StockTwits are labeled. To increase the accuracy of stock price prediction, we need a powerful method for the sentiment analysis of top authors. Deep Learning is beneficial in facing a large amount of unsupervised data (Big Data) like data provided in social media. In our paper, we adopt Deep Learning to do sentiment analysis of top authors. We believe that using Deep Learning can vastly improve correct classification in sentiment analysis regarding various stock picks and thus exceed the current accuracy of stock price prediction.

### **Big Data**

The term Big Data has been in use since the 1990s. In 2012 Gartner update his previous definition regarding Big Data and defines it as follows: “Big Data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision-making, and process automation”. Big Data is referred to the growing digital data that are difficult to manage and analyze using traditional software tools and technologies. Big Data often has a large number of samples, a large number of class labels and very high dimensionality (attributes). The target size of the Big Data moving continually in 2012 was ranging around a few dozen terabytes to many petabytes of data. There are four attributes including volume, variety, velocity, and veracity that define Big Data [56] Obviously, data volume is the primary attribute of Big Data. By increasing the volume of the Big Data, the complexity, and the underneath relationships of data increased as well. Raw data in a Big Data system is unsupervised and diverse although it can consist a small quantity of supervised data. Many social media companies including Facebook, Twitter, StockTwits, LinkedIn have a large amount of data. As data become bigger Deep Learning approach become more important to provide Big Data analysis.

The other thing that makes Big Data really big is the variety of data. Big Data coming from a variety of sources than ever before. Web sources including social media, click-streams, and logs are some example of these resources. One of the challenges in Big Data processing is working with a Variety of different data. In order to extract a structured representation of data, Big Data needs to do preprocessing on unstructured data.

Velocity is another feature of Big Data. The frequency of data generation in Big Data is fast. For example, consider the stream of message coming from StockTwits website. Velocity is just as important as the volume and variety characteristics of Big Data. The quickness of processing input into usable information is important to deal with velocity associated with Big Data.

Veracity refers to the trustworthiness of the data in Big Data. By increasing the number of data sources and types trust in Big Data become a practical challenge. In addition to the four vs. there are lots of challenges including data cleansing, feature engineering [57–59], high-dimensionality, and data redundancy that Big Data analytics face. Deep Learning is used in industrial products that have the opportunity to have a large volume of digital data (Big Data). Google uses Deep Learning algorithms and Big Data available on the Internet for Google’s translator. In some Big Data application domains such as social media, marketing, and financial data feeds using Deep Learning algorithms and architecture for analyzing large-scale [60, 61], fast-moving streaming data

is encouraged, but still analyzing Big Data by using Deep Learning application remain unexplored.

In Big Data environments, it is critical to analyze, decide and act quickly and often. Big Data has the potential to make a huge change in science and all aspects of our society, but extracting information from Big Data is not an easy task. Decentralized control and autonomous data sources are two other important characteristics of Big Data. Each data source can collect information without any centralized control [62]. Big Data technology is still young, there are many technical problems in stream computing, parallel computing, Big Data architecture, Big Data model, and software systems that can support Big Data, etc should be investigated.

Today, machine learning techniques especially Deep Learning models, together with powerful computers play an important role in Big Data analysis. Deep Learning methods can leverage the predictive power of Big Data in fields like search engines, medicine, and astronomy. In contrast to conventional datasets used for data mining approach which was noise free, Big Data is often incomplete because of their disparate origins. Big Data brings transformative potential and big opportunities for various fields. Typical data mining algorithms require having all data in main memory this is a clear technical difficulty for Big Data which is spread across different locations. In addition, data mining methods need to overcome sparsity, heterogeneity, uncertainty, and incompleteness of Big Data as well. Deep Learning and Big Data are considered as the big deals and the bases for an American innovation and economic revolution. Even in government and society Big Data emerge as a useful remedy to solve some problems. In 2012 the Obama Administration announced a “Big Data research and development initiative” to help solve some of the Nation’s most pressing challenges.

### **Sentiment analysis**

Following the early work in sentiment analysis done in [63, 64], we examine source materials and apply natural language processing techniques to determine the attitude of the writer towards a subject. Generally speaking, sentiment analysis is a form of classifying text documents to numerous groups. Most of the time, we need only to classify documents into positive and negative classes [65]. Furthermore, there are different methods in sentiment analysis that can help us to measure sentiments. These methods include lexical-based approaches methods and supervised machine learning methods. Machine learning models are more popular because lexical-based approaches, which are based on the semantics of words, use a predefined list of positive and negative words to extract the sentiment of new documents. Creating these predefined lists is time-consuming and we cannot build a unique lexical-based dictionary to be used in every separate context. With the growing popularity of social media, huge datasets (Big Data) of reviews, blogs, and social network feeds are being generated continuously. Big Data techniques are used in application domains that we collect and maintain a massive amount of data. Growing data, intensive technologies, and increasing data storage resources develop Big Data science. The main concept in Big Data analytics is extracting a meaningful pattern from a huge amount of data. Big Data need special methods that can be used to extract patterns from a massive amount of data. Deep Learning has this opportunity to provide a solution to address the learning and data analysis problem that exists in a massive amount

**Table 1 Performance of the logistic regression on the StockTwits dataset**

Accuracy	Precision	Recall	F-measure	AUC
0.7088	0.7134	0.6980	0.7056	0.7088

of data (Big Data) and also they are better at learning complex data patterns. There are other Big Data problems such as domain adoption and streaming data that large-scale Deep Learning models for Big Data analytics have to contend with them. Concepts and methods from sentiment analysis that can help us to extract information from these areas have become increasingly important as businesses, organizations, and individuals seek to make better use of their Big Data. In the following section, we start our investigation the performance of sentiment analysis based on data mining approaches for our dataset.

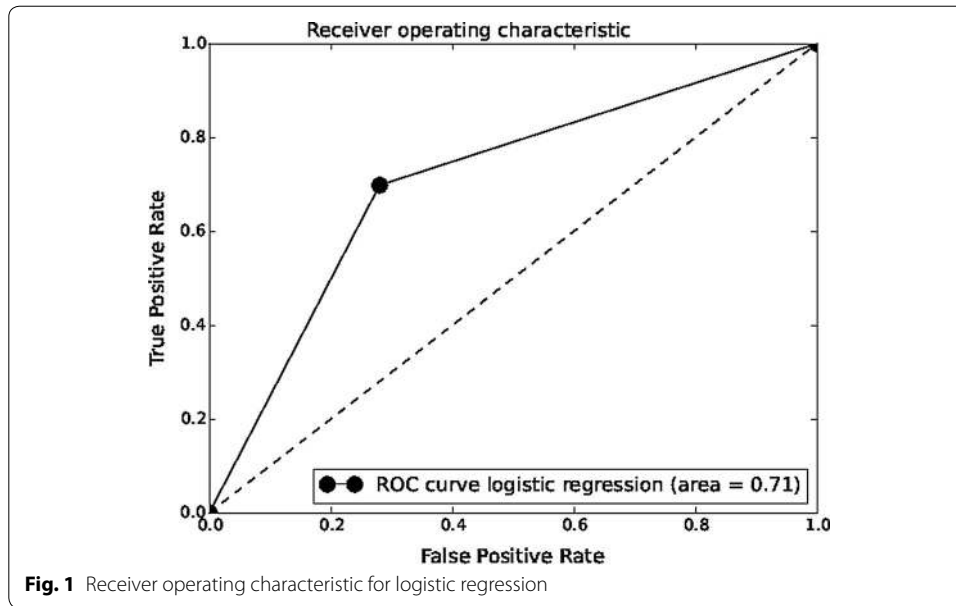
#### Sentiment analysis with data mining approaches

Wang in [2] uses a supervised data mining approach to find the sentiment of messages in the StockTwits dataset. They removed all stopwords, stock symbols, and company names from the messages. They consider ground-truth messages as training data and test multiple data mining models, including Naïve Bayes, Support Vector Machines (SVM), and Decision Trees. By running tenfold cross validation, they found that the SVM model produces the highest accuracy (76.2%). They used unigrams as features and removed infrequent unigrams that occur less than 300 times over all messages because using n-grams can lead to data sparsity problem. As a result, it is necessary to use lower-order n-grams to address sparsity problem otherwise performance would be decreased. On the other hand, by using lower-order n-grams we lose the order of the words in a sentence. As we know the order of the words in a sentence can help us to better understanding the sentiment of a document. We believe that using Deep Learning to predict sentiment of authors can help us to overcome these problems and increase the accuracy of prediction. Deep Learning nonlinear feature extraction can improve data mining results and classification modeling [55]. Logistic regression uses the logistic sigmoid function to weighted input values to classify input data, it is similar to a Deep Learning without hidden layers. Logistic regression is used as a classifier in the final layer of a Deep Learning. In other words, Deep Learning algorithms work as multiple feature learning steps. Logistic regression is very fast and simple so it is used for large datasets.

We follow Wang [2] approach and apply logistic regression [3] on the StockTwits dataset. In Table 1, we provide the performance of logistic regression on StockTwits data based on different performance metrics. Also in Fig. 1, we present the ROC curve [66] for this model.

#### Increase accuracy by using feature selection

One of the problems that prevent us from accurately classifying a Big Data is the noise found within it. Feature selection, including the removal of noisy features and elimination of ineffective vocabulary, makes training and applying a classifier more effective [67]. The existing approaches to finding an adequate subset of features fall into two groups: feature filters and feature wrappers [68]. In feature filters, the final set of features is



**Fig. 1** Receiver operating characteristic for logistic regression

selected based on the statistical properties of those features. With feature wrappers, an iterative search process is applied through a modeling tool's results. In each iteration, a candidate set of features is used in the modeling tool and the results are recorded. Each step uses the results from the previous step, and so new tentative sets are generated. This process is repeated until some specified convergence criteria are met. In our experiment, we had a huge number of features and instances, and thus, our data was very sparse. We tried several feature selection methods to see how they would affect the accuracy of our sentiment analysis.

From the methods tested, we selected three feature filters which included Chi-squared, ANOVA, and mutual information. The advantages of using these feature selection techniques are their speed, scalability and their independence of the classification. Our reasoning for choosing these methods is their ability to deal with sparse data. On the other hand, these methods have some drawbacks as well, they ignore feature dependencies and also they ignore interaction with the classifier [69]. In this section, we examine these methods and the results of applying them to our dataset.

### Chi-square

Pearson's Chi-squared [70] test is used for two types of comparison: a test of independence or a test of goodness of fit. We apply the test of independence to our dataset to see if the occurrence of a specific feature is independent of the class. Our terms are ranked by their score as determined with Eq. (1). In this equation, 'O' stands for observed frequency, and 'E' stands for expected frequency. A high  $X^2$  score rejects the null hypothesis of independence of the term and class.

$$X^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i} \quad (1)$$

**Table 2 Performance of the Chi-squared feature selection on the StockTwits dataset**

Features	Accuracy	Precision	Recall	F-measure	AUC
55,820	0.7088	0.7134	0.6980	0.7056	0.7088
40,000	0.4796	0.4851	0.6645	0.5608	0.4796
20,000	0.5018	0.5013	0.6879	0.5800	0.5018
4000	0.5274	0.5206	0.6946	0.5951	0.5274
2000	0.5221	0.5190	0.6036	0.5581	0.5221
400	0.5308	0.5278	0.5834	0.5542	0.5308
200	0.5333	0.5280	0.6284	0.5738	0.5333
50	0.5314	0.5232	0.7071	0.6014	0.5314

Applying Chi-squared to our dataset and decreasing the number of features gradually allowed us to see how it can affect the performance of logistic regression. Classifier results are provided in Table 2. Reducing the number of features increases accuracy in some cases—for example, by reducing the number of features from 40,000 to 500, accuracy increases by seven percent. However, this is an irregularity in our dataset and does not mean that Chi-squared is an effective feature selection method to increase the accuracy of our classifier.

#### Analysis of variance

One of the other feature selection methods that we used was the analysis of variance (ANOVA) feature selection. ANOVA [71] is used to determine if there are any statistically significant differences between the arithmetic means of independent groups. By using ANOVA for feature selection in our experiment, we clarify the relevance of terms by assigning a score to each based on an F-test. Top scoring terms are considered as our desired features and sent to the classification models.

The F-test formula is shown in Eq. (2).

$$F = \frac{MS_B}{MS_W} \quad (2)$$

In this equation  $MS_B$  is *between-group variability* Eq. (3), and  $MS_W$  is *within-group variability* Eq. (4). In between-group variability  $n_i$  is the total number of observations of class  $i$ ,  $m$  is the number of classes and  $\bar{x}$  denotes the general mean of the data.

$$MS_B = \frac{\sum_i n_i (\bar{x}_i - \bar{x})^2}{m - 1} \quad (3)$$

In within-group variability,  $x_{ij}$  denotes the  $j$ th observation in the  $i$ th class [72].

$$MS_W = \frac{\sum_{ij} (x_{ij} - \bar{x}_i)^2}{n - m} \quad (4)$$

By extracting more effective features based on F-test scores, we examined whether ANOVA feature selection improves the accuracy of the classification methods. Per the results provided in Table 3, accuracy is not improved through ANOVA feature selection, so it will not be used for further testing.

**Table 3 Performance of the ANOVA F-test feature selection on the StockTwits dataset**

Features	Accuracy	Precision	Recall	F-measure	AUC
55,820	0.7088	0.7134	0.6980	0.7056	0.7088
40,000	0.7094	0.7130	0.7010	0.7070	0.7094
20,000	0.7091	0.7127	0.7007	0.7066	0.7091
4000	0.5274	0.5206	0.6946	0.5951	0.5274
2000	0.7045	0.7048	0.7038	0.7043	0.7045
400	0.6785	0.6638	0.7233	0.6923	0.6785
200	0.6611	0.6378	0.7457	0.6875	0.6611
50	0.6191	0.5863	0.8084	0.6797	0.6191

**Information gain**

Our results show that ANOVA and Chi-square feature selection methods cannot considerably increase the accuracy of our classification models. In this section, we look at mutual information feature selection, which is one of the most commonly used feature selection methods. Mutual information is defined as the number of dependencies between two random variables. This allows us to determine information gain, which is the amount of information acquired about one random variable through another random variable. Mutual information between two random variables ( $X$  and  $Y$ ) is defined in Eq. (5).

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right) \quad (5)$$

In this equation, if  $x$  and  $y$  are independent, i.e. ( $p(x, y) = p(x) \times p(y)$ ), their mutual information will be zero. Which, in turn, means that by knowing one of these random variables we cannot gain any information about the other one.

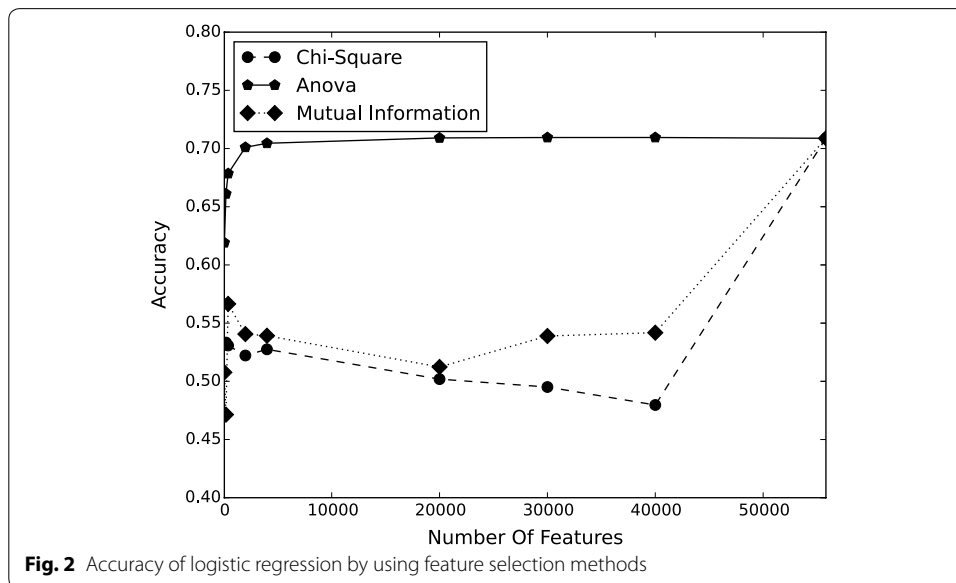
By using mutual information for feature selection, we explore how much information each term provides to making the correct classification decision. This method extracts features with the highest mutual information value. In this way, we will have features that contain the most information about the class. In our experiment, mutual information for feature selection was also not effective. This is shown by the results provided in Table 4.

In Fig. 2 we decrease the number of features by applying selection methods containing Chi-square, ANOVA, and information gain and compare the accuracy of logistic regression. As our results demonstrate, feature selection methods cannot considerably improve the accuracy of logistic regression. Data mining algorithms cannot extract the complex and nonlinear patterns that exist in Big Data. Extracting these features, Deep Learning can use simpler linear models for Big Data analysis tasks including classification and prediction which is important when we deal with the scale of Big Data.

With the result of logistic regression based on the bag-of-words model used as a baseline, we investigate whether Deep Learning methods can improve the accuracy of this logistic regression in Big Data. The bag-of-words model does not consider word order and other words in a sentence, and it has a limited sense of word sentiment. We believe

**Table 4** Performance of the mutual information feature selection on the StockTwits data-set

Features	Accuracy	Precision	Recall	F-measure	AUC
55,820	0.7088	0.7134	0.6980	0.7056	0.7088
40,000	0.5417	0.5311	0.7115	0.6082	0.5417
20,000	0.5123	0.5087	0.7144	0.5943	0.5123
4000	0.5391	0.5337	0.6190	0.5732	0.5391
2000	0.5406	0.5350	0.6193	0.5741	0.5406
400	0.5665	0.5540	0.6815	0.6112	0.5665
200	0.4713	0.4760	0.5692	0.6126	0.5459
50	0.5077	0.5052	0.7414	0.6009	0.5077



that using Deep Learning methods instead of the bag-of-words may help us to improve the accuracy of our model. In consecutive layers of deep architectures in Deep Learning, each layer applies a nonlinear transformation on its input and provides a representation of its output. On the other word, Deep Learning can learn representations of the Big Data in a Deep Architecture with multiple levels of representations. It is important to consider that transformations in the layers of Deep Learning are nonlinear and try to extract underlying factors in the Big Data. The output of final layer (the final representation of data which constructed by Deep Learning algorithm) can be used as features for classifiers or other applications. In our paper, we mainly focus to see how Deep Learning can assist with sentiment analysis in StockTwits data and which Deep Learning algorithm can be adapted to improve the accuracy of sentiment analysis in StockTwits in compare to data mining models. With respect to the first topic, we explore three Deep Learning algorithms including doc2vec [73–75], LSTM [76] and CNN [77] to see if they can more accurately predict StockTwit’s users’ sentiment.

### Deep Learning in Big Data analytics

In this section, we explore advantages of using Deep Learning algorithms in Big Data analysis. Also, we take a look at some Big Data characteristics that challenges Deep Learning in Big Data analysis.

Big Data analytics, provide the opportunity to develop novel algorithms to address some issues related to Big Data. Deep Learning algorithms are one of these solutions. For instance, the representations extracted by Deep Learning can be used in Big Data analytics approach. In addition, when Big Data is represented in a higher form of abstraction, linear modeling can be considered for Big Data analytics. There are various works that have been performed by using Deep Learning algorithms.

Deep Learning dates back to the 1940s. Its only appears to be new, because Deep Learning was relatively unpopular for several years preceding its current popularity, and because Deep Learning has gone through many different names, only recently being called “deep learning.” Deep Learning has been rebranded many times, reflecting the influence of different researchers and different perspectives. Some basic context of the history of Deep Learning is useful for understanding Deep Learning. Deep Learning is known as cybernetics in the 1940s–1960s, Deep Learning known as connectionism in the 1980s–1990s, and the current resurgence under the name Deep Learning beginning in 2006 [78].

As we mentioned before, Deep Learning algorithms extract an abstract representation of Big Data through multi-level hierarchical learning. Deep Learning is attractive for extracting information from Big Data because it can be used to learn from a massive amount of unlabeled data. Once Deep Learning learned unsupervised data (Big Data) more traditional models can be trained with less amount of labeled data [79–81]. Global relationships in the Big Data can perform better by using Deep Learning.

Some of the advantages of learned abstract representations by Deep Learning include, a simple model can work effectively with the knowledge of more abstract data representation, automation of data representation extraction can lead to a broad application to different data types. These specific characteristics of Deep Learning make it desirable for Big Data analytics.

Deep Learning algorithms can be used to address the problem of volume and variety of Big Data analytics. Effectively using a massive amount of data (volume in Big Data) is one of the advantages of Deep Learning. Since Deep Learning deals with data abstraction it is desirable to work with raw data in different formats and resources (variety in Big Data) and minimize a need for feature selection from new data type observed in Big Data.

However Big Data has some characteristics including streaming and fast moving which can lead to some challenges for adopting Deep Learning for Big Data. Deep Learning needs to be adapted to lead with a lot of continuous Big Data. There are some works associated with Deep Learning and streaming Big Data. For instance, adoptive deep belief networks introduced in [82] illustrate how Deep Learning can be used to learn from streaming data. Zhou et al. [83] describe how Deep Learning algorithms can be used for feature learning on Big Data. One of the other problem that associate of using Deep Learning in Big Data is using Deep Learning for large-scale models and massive datasets. In [84] Dean uses thousands of CPU cores to train a Deep Learning neural

network with billions of parameters. Coates et al. [85] suggest using the power of a cluster of GPU servers to overcome the problem of Deep Learning in large-scale datasets.

Big Data encompasses a lot of things from medicine, genomic and biological data to call center. To handle huge volumes of input associated with Big Data, large-scale Deep Learning models are desirable. They can illustrate the optimal number of model parameters and overcome the challenges of Deep Learning for Big Data analysis. There are other Big Data problems like domain adaption and streaming data that large-scale Deep Learning models for Big Data need to handle.

Variety is one of the other characteristics of Big Data, which focuses on the variation of the input domains and data types in Big Data so the problem of domain adoption is another issue that Deep Learning in Big Data analysis need to overcome. There are some studies including [86, 87] that mainly focus on domain adoption during the learning process. Glorot et al. [86] illustrate that Deep Learning can find intermediate data representations in a hierarchical learning manner and this representation can be used for other domains. Chopra et al. [87] propose a new Deep Learning model for domain adoption. Their new proposed Deep Learning model considers information available from the distribution shift between the train and test data. Our paper mainly focuses on information retrieval so in the following section, we summarize Deep Learning in sentiment analysis.

### **Sentiment analysis with Deep Learning approaches**

In the prior section, we discussed some advantages of using Deep Learning in Big Data analysis including the application of Deep Learning algorithms for Big Data analysis and how specific characteristics of Big Data can lead to some challenges in adopting Deep Learning algorithms for Big Data analytics tasks. In this section, we explore sentiment analysis using Deep Learning algorithms. In data mining prediction tasks feature engineering is the most important and most difficult skill. The effort involved in feature engineering is the main reason to seek algorithms that can learn features by themselves. Hierarchical feature learning in Deep Learning extracts multiple layers of non-linear features and then a classifier combines all the features to make predictions [88]. Data mining models based on shallow learning like Support Vector Machines and Decision Trees are not able to extract complex features. On the other hand, Deep Learning algorithms have the capability to generalize in global ways, generating learning patterns, and relationships beyond immediate neighbors in the Big Data [79]. In order to gain more complex features, Deep Learning algorithms transform first features like edge and blobs in image again to extract more informative features to distinguish between classes. This process is very close to brain activity. The first hierarchy of neurons which are sensitive to specific edges and blobs receive information in the visual cortex [89] while brain regions further down the visual pipeline are sensitive to more complex structures such as faces. So in other words, Deep Learning learns the representation of Big Data in a deep architecture and more layers the data goes through, the more complicated the non-linear transformations which are constructed. But hierarchical feature learning suffered from major problems such as the vanishing gradient for very deep layers, this problem makes these architectures perform poorly in comparison to shallow learning algorithms. Deep Learning methods can overcome vanishing gradient problem so they can train with dozens of layers of non-linear hierarchical features. Not only Deep Learning

methods are related to learning deep non-linear hierarchical features they can also be used to detect very long non-linear time dependencies in sequential data. Long short-term memory (LSTM) and Recurrent Neural Networks are two examples of neural networks that can increase the accuracy of prediction by picking up on activity hundreds of time steps in the past. One of the main problems in Big Data is storing data effectively and retrieve information from this Big Data. Deep Learning algorithms can be used to generate high-level abstract data representation which will be used for sentiment analysis (especially for raw Big Data input). While a vector representation of Big Data provides faster information retrieval, Deep Learning can be used for relational understanding of the Big Data. Using Deep Learning algorithms can help us to extract semantic features from a massive amount of text data in addition to reduce dimensions of the data representations. Hinton et al. [90, 91] propose a Deep Learning model to learn the binary codes for documents. The word count vector of a document is the lowest layer and the learned binary code of the documents is the highest layer. The binary code can be used for information retrieval in Big Data. we can use some unsupervised data in training a Deep Learning model [92, 93]. Ranzato et al. [94] propose a study in which Deep Learning model learn with supervised and unsupervised Big Data. Deep Learning algorithms provide this opportunity to extract semantic aspect of a document by capture complex nonlinear representations between word occurrences. Using Deep Learning can help us leverage unlabeled document (unsupervised Big Data) to have access to huge amount of data (Big Data). Unlabeled data are often ambled and cheap to collect in Big Data. Since Deep Learning relatively recently becomes popular, additional work needs to be done to use hierarchical learning strategy as a method for sentiment analysis of Big Data.

### **word2vec**

Mikolov in [95], proposed word2vec model. In this model, instead of relying on the number of occurrences of the words, neural network methods (Deep Learning) are used to produce a high-dimensional vector representation of each word or document. Word2vec uses the location of words relevant to each other in a sentence to find the semantic relationship between them. In contrast to the bag-of-words model, word2vec can capture sentimental similarity among words.

Word2vec is implemented in two different model architectures, *continuous bag-of-words* and *skip-gram*. In the continuous bag-of-words architecture, we have a sequence of words and we need to predict which word is more likely to be the next word in this sequence. In the skip-gram architecture, with each word, we try to find a more probabilistic surrounding window of words. The outcome is in a vector space, words with semantic similarity are nearby. When using the word2vec model, the order of the words in a sentence is ignored, and only words and their distance from each other are considered. Le and Mikolov [73], describe the doc2vec method. doc2vec generalizes word2vec by adding a paragraph vector. This inclusion means that each paragraph, like each word, is mapped to a vector. The advantage of considering a paragraph as a vector is that it can work as a kind of memory to keep the order of the words in a sentence.

Doc2vec, like word2vec, is implemented in two different methods *distributed memory* and *distributed bag-of-words*. In distributed memory, a paragraph is treated the same as a word. This is word2vec beneficial because after paragraph vectors have been learned



a common problem in Deep Learning. Because of the vanishing gradient problem, RNNs look back just a few steps. Although vanishing gradients are not exclusive to RNNs, they limit our network depth to less than the length of the sentence. Thankfully, there are a variety of methods that can help us address the vanishing gradient problem. For example, instead of using *tanh* or sigmoid as activation functions, we can use ReLU. However, we chose a more popular solution for our work—*Long short-term memory (LSTM)*.

LSTM was proposed [76] by Hochreiter and Schmidhuber. The main difference between RNNs and LSTMs is the gated cell. Gated cells in LSTMs help the system store more information in comparison to RNNs. Information can be stored in, written to, or read from a cell. Cells decide whether to remove or store information by opening and closing gates. A cell is composed of four main elements: an input gate, a neuron with a self-recurrent connection, a forget gate, and an output gate. The forget gate is an element which allows the cell to remember or forget its previous state. For example, assume that we want to capture the gender of the subject. In this case, when seeing a new subject, the previous one should be forgotten so that a relevant information can be determined and stored.

### ***Convolutional neural network***

One of the most commonly used Deep Learning models is the fully-connected neural network. Although fully-connected neural networks are considered as a good solution in classification tasks, the huge number of connections in these networks may lead to problems. These problems can be further amplified in text processing because of the high number of neurons required. In addition, we believe that words which are close together in a sentence are more to each other when compared to words which never appear close together in any sentence. But fully-connected neural networks treat input words which are far apart the same as words which are close together in a sentence. The hierarchical learning process of Deep Learning makes it expensive for high-dimensional data like image or text. On the other words, these kinds of Deep Learning algorithm can be stalled when dealing with Big Data that shows large Volume (one of the features of Big Data).

Convolutional neural networks offer certain advantages that make them desirable to address these problems. First, each neuron in the first hidden layer, instead of connecting to all input neurons, is only connected to a small region of them. This reduction in connection complexity works to also reduce potential computational problems. Second, using the same weights for each of the hidden neurons provides the opportunity to detect the same feature in different locations in the input text. At the end of the network, a pooling layer simplifies the information from the convolutional layers to the output [97]. The convolutional neural network is one of the methods that can be used effectively for Big Data analysis. The convolutional neural network which is one of the powerful models in Deep Learning, use convolutional layers to filter inputs for useful information.

Hinton et al. [29] use a Deep Learning and convolutional neural network for image object recognition. Their Deep Learning model outperforms other existing approaches. Hinton's team work is valuable because they show the importance of Deep Learning in image searching. Dean et al. in [84] use similar Deep Learning modeling approach but

**Table 5 Performance of doc2vec on the StockTwits dataset**

Window	Accuracy	Precision	Recall	F-measure	AUC
5	0.6202	0.6097	0.6682	0.6376	0.6202
10	0.6723	0.6687	0.6830	0.6757	0.6723

with a large-scale software infrastructure as a training and in [98] video data is used. They use Deep Learning method like stacking and convolution to learn hierarchical representation.

### Results and discussion

In this section, we will explain our experiments in applying Deep Learning methods on the StockTwits dataset. We tried to see if Deep Learning models could improve the accuracy of sentiment analysis of StockTwits messages. Deep Learning attempt to mimic the hierarchical learning approach of the human brain. Using Deep Learning in extract features bring non-linearity to the Big Data analysis. The results of applying three commonly used Deep Learning methods in natural language processing are provided in the following section.

#### *Doc2vec*

As our first step, we apply the doc2vec model to the StockTwits dataset to see if it can increase the accuracy of sentiment prediction for stock market writers. This was chosen as the first model because it uses the paragraph as a memory to keep the order of the words in a sentence, and maps paragraphs, as well as words, to a vector.

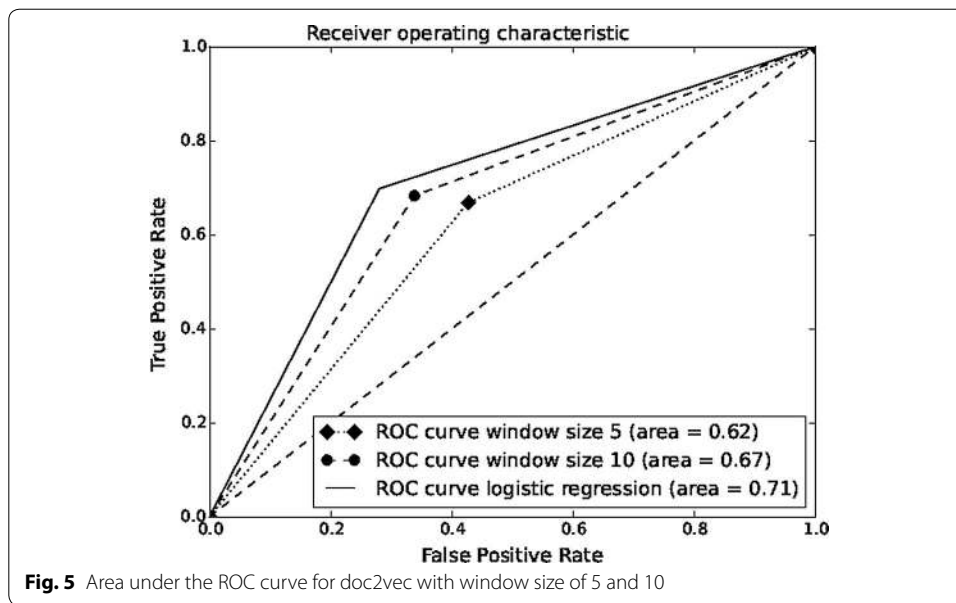
Le in [73] recommends using both doc2vec architectures simultaneously to create a paragraph vector. Following his method in our experiment, each paragraph vector is a combination of two vectors—one learned by distributed memory architecture (DM) and the other learned with distributed bag-of-words (DBOW) architecture. The accuracy of the doc2vec model is also likely to be affected by window size; with larger windows having higher accuracy. In order to evaluate this, we consider windows of the most commonly-used sizes—5 and 10. The Gensim library in Python was used to implement doc2vec and all words with a total frequency of less than two were ignored. The results are shown in Table 5.

As we expected, the accuracy of applying doc2vec for a window size of 10 is higher than with a window size of 5, but their difference is negligible.

By comparing the results of applying logistic regression as a baseline on the StockTwits dataset in Table 1 with the results of doc2vec in Table 5, we find that doc2vec cannot be an effective model to predict sentiment in the StockTwits dataset. In Fig. 5 we provide the receiver operating characteristic curve for the window sizes of five and ten and compare their results with the ROC of the logistic regression.

#### *Long short-term memory*

Based on the findings in the previous section, doc2vec is not a good model for predicting sentiment of authors regarding the stock market, and so we move on to RNNs. These are some of the other most popular models for use in Natural Language processing



**Table 6** Performance of the LSTM on the StockTwits dataset

Accuracy	Precision	Recall	F-measure	AUC
0.6923	0.8518	0.6571	0.7419	0.7109

have shown very good results. RNNs were adopted to see if they can help improve the accuracy of StockTwits sentiment analysis. Although an actual RNN was not used for our experiment, LSTM [99–102] could be a viable replacement because it has a deeper memory structure.

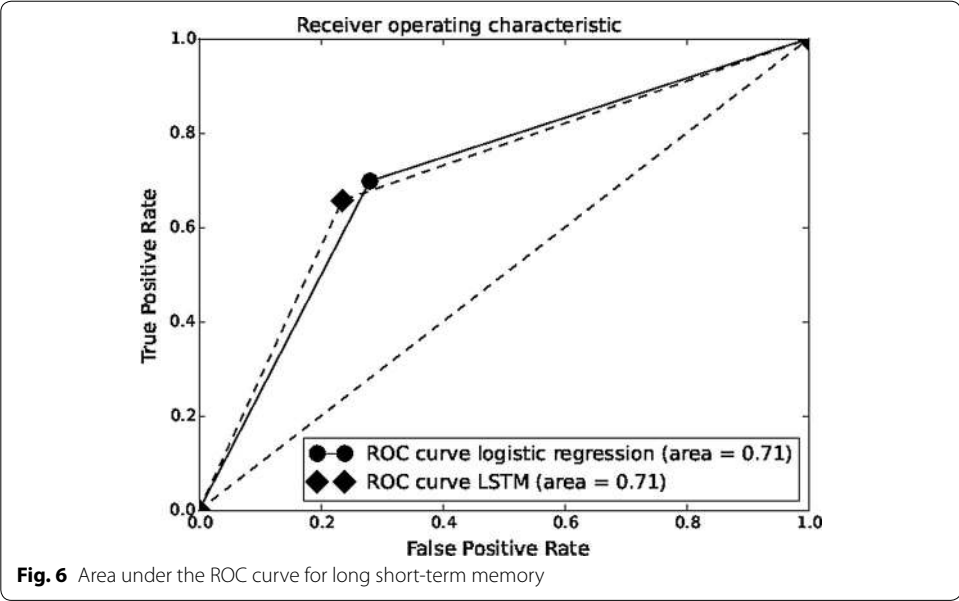
In our implementation, we used the Theano [103] library in Python. We use average pooling as our pooling method. For the last step, we fed the result of the pooling to a logistic regression layer to find the target class label associated with the current input sequence. We present the result of our experiments in Table 6. Although using LSTM compared to doc2vec did increase our accuracy, it is still lower than our requirements.

Using logistic regression as the baseline and comparing results in Tables 1 and 6 reveals that LSTM is not an effective model for predicting sentiment in the StockTwits dataset. In Fig. 6, we compare the area under the ROC curve for the results of applying LSTM and logistic regression.

### Convolutional neural network

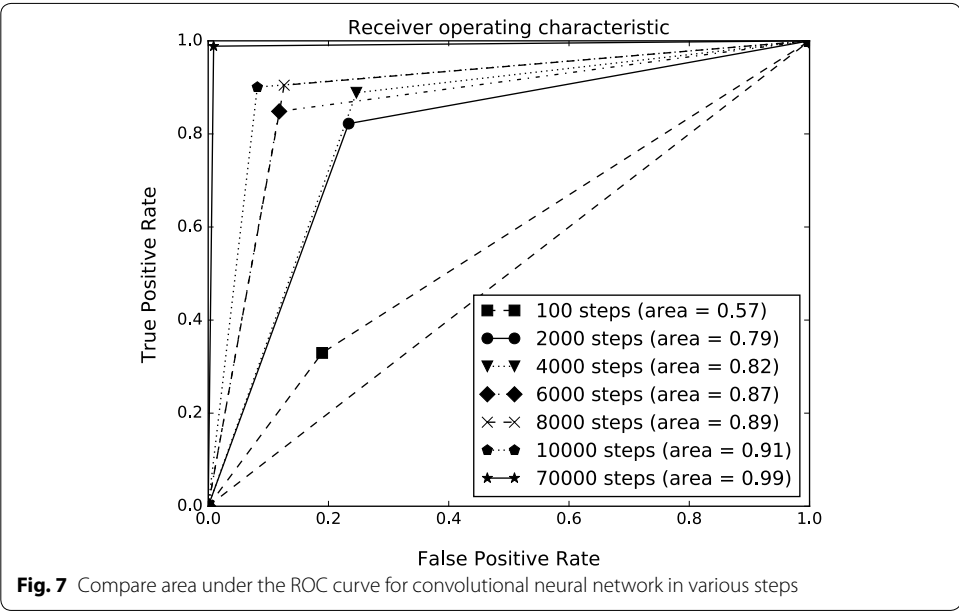
With LSTM being found ineffective, we turn to the CNN. Although CNN is very popular in image processing, the ability to find the internal structures of a Big Data makes it a desirable model for our purposes. We employ CNN to see if it can be used to improve our sentiment analysis task by using the Tensorflow [104] package in Python. The first step of our process is embedding words into low dimensional vectors.

After that, we perform convolutions with different filter sizes over the embedded word vectors. In our experiment, we used filter sizes of 3, 4 and 5. Then we apply max pooling



**Table 7** Performance of the convolutional neural network on the StockTwits dataset

Steps	Accuracy	Precision	Recall	F-measure	AUC
100	0.5700	0.6348	0.3294	0.4338	0.5700
2000	0.7943	0.7787	0.8221	0.7999	0.7943
4000	0.8210	0.7828	0.8885	0.8323	0.8210
6000	0.8651	0.8778	0.8484	0.8629	0.8651
8000	0.8891	0.8774	0.9046	0.8908	0.8891
10,000	0.9093	0.9168	0.9004	0.9086	0.9093
70,000	0.9897	0.9909	0.9885	0.9897	0.9897



**Table 8 Compare Deep Learning models in financial sentiment analysis**

Model	Accuracy	Precision	Recall	F-measure	AUC
Logistic regression	0.7088	0.7134	0.6980	0.7056	0.7088
Doc2vec	0.6723	0.6687	0.6830	0.6757	0.6723
LSTM	0.6923	0.8515	0.6571	0.7419	0.7109
CNN (10,000 steps)	0.9093	0.9168	0.9004	0.9086	0.9093

on the result of the convolution and add dropout regularization. The process concludes by using a softmax layer to classify our results.

Table 7 shows the results of these operations. By comparing the accuracy of logistic regression as a baseline in Table 1 with the results of applying convolutional neural network provided in Table 7, we conclude that CNN outperforms logistic regression after less than 2000 steps. After 6000 steps the accuracy of CNN is around 86% which is considerably higher than the other models. Additionally in Fig. 7, we provide the receiver operating characteristic curve for CNN, which compares the area under the roc curve after applying CNN in multiple steps. As evident in Table 7, with proceeding steps in CNN, the ROC curve gets closer to the top left corner of the diagram. This proves that by proceeding stepwise in CNN on the StockTwits dataset, the accuracy of prediction increases gradually. We compare the result of logistic regression, doc2vec, LSTM, and CNN (after 10,000 steps) in Table 8. Based on the results, we find that CNN is an effective model for predicting the sentiment of authors in the StockTwits dataset as it outperformed all other models in all five performance measurement.

## Conclusions

Deep Learning has good performance and promise in many areas, such as natural language processing. Deep Learning has this opportunity to address the data analysis and learning problems in Big Data. In contrast to data mining approaches with its shallow learning process, Deep Learning algorithms transform inputs through more layers. Hidden layers in Deep Learning are generally used to extract features or data representations. This hierarchical learning process in Deep Learning provides the opportunity to find word semantics and relations. These attributes make Deep Learning one of the most desirable models for sentiment analysis.

In this paper, based on our results we show that convolutional neural networks can overcome data mining approach in stock sentiment analysis. In standard data mining approach to text categorization, documents represent as bag-of-word vectors. These vectors represent which words appear in a document but do not consider the order of the words in a sentence. It is clear that in some cases, the word order can change the sentiment of a sentence. One remedy to this problem is using bi-grams or n-gram in addition to uni-gram [86, 105, 106]. Unfortunately, using n-grams with  $n > 1$  is not effective [107]. Using CNN provides this opportunity to use n-grams to extract the sentiment of a document effectively. It benefits from the internal structure of data that exists in a document through convolution layers, where each computation unit responds to a small region of input data. We used logistic regression, which works based on a bag-of-words, as a baseline and compared the result of applying Deep Learning to logistic

regression. Based on our results, among different common Deep Learning methods in sentiment analysis, only convolutional neural network outperforms logistic regression. The accuracy of convolutional neural networks, in comparison to the other models, is considerably better. Based on our results we can use CNN to extract the sentiment of authors regarding stocks from their words. There are some people in the financial social network who can correctly predict the stock market. By using CNN to predict their sentiment we can predict future market movement.

#### Authors' contributions

SS is the main author, she did all the research and explored different methods. The idea of doing sentiment analysis and classifying users to bullish and bearish is DW idea. We consult with AP and TMK. AP helps us in financial concepts and we use TMK experience to improve our work. All authors read and approved the final manuscript. All authors read and approved the final manuscript.

#### Author details

<sup>1</sup> Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA. <sup>2</sup> College of Business, Florida Atlantic University, Boca Raton, FL 33431, USA.

#### Acknowledgements

Not applicable.

#### Competing interests

The authors declare that they have no competing interests.

#### Availability of data and materials

Not applicable.

#### Consent for publication

Not applicable.

#### Ethics approval and consent to participate

Not applicable.

#### Funding

Not applicable.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 20 April 2017 Accepted: 28 December 2017

Published online: 25 January 2018

#### References

1. Ellison NB, et al. Social network sites: definition, history, and scholarship. *J Comput Mediat Commun*. 2007;13(1):210–30.
2. Wang G, Wang T, Wang B, Sambasivan D, Zhang Z, Zheng H, Zhao BY. Crowds on wall street: extracting value from social investing platforms, foundations and trends in information retrieval. New York: ACM; 2014.
3. Freedman DA. Statistical models: theory and practice. Cambridge: Cambridge University Press; 2009.
4. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–105.
5. Socher R, Huang EH, Pennin J, Manning CD, Ng AY. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Adv Neural Inf Process Syst*. 2011;24:801–9.
6. Pearson K. Notes on regression and inheritance in the case of two parents. *Proc R Soc Lond*. 1895;58:240–2.
7. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP); 2013.
8. Dahl G, Mohamed AR, Hinton GE. Phone recognition with the mean–covariance restricted Boltzmann machine. *Adv Neural Inf Process Syst*. 2010;23:469–77.
9. George E, Yu D, Deng L, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans Audio Speech Lang Process*. 2012;20(1):30–42.
10. Seide F, Li G, Yu D. Conversational speech transcription using context-dependent deep neural networks. In: Twelfth annual conference of the international speech communication association; 2011.
11. Mohamed A, Dahl GE, Hinton G. Acoustic modeling using deep belief networks. *IEEE Trans Audio Speech Lang Process*. 2012;20(1):14–22.
12. Itamar A, Rose DC, Karnowski TP. Deep machine learning—a new frontier in artificial intelligence research [research frontier]. *IEEE Comput Intell Mag*. 2010;5(4):13–8.

13. Najafabadi NM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. *J Big Data*. 2015;2:1.
14. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278–324.
15. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. *J Mach Learn Res*. 2011;12:2493–537.
16. Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: *Proceedings of the 25th international conference on machine learning*. London: ACM; 2008. p. 160–7.
17. Gao J, Deng L, Gamon M, He X, Pantel P. Modeling interestingness with deep neural networks. 2014. US Patent App. 14/304,863.
18. Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*. 2014.
19. Kim Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*. 2014.
20. Shen Y, He X, Gao J, Deng L, Mesnil G. A latent semantic model with convolutional-pooling structure for information retrieval. In: *Proceedings of the 23rd ACM international conference on information and knowledge management*. New York: ACM; 2014. p. 101–10.
21. Liheng X, Liu K, Lai S, Zhao J, et al. Product feature mining: semantic clues versus syntactic constituents. *ACL*. 2014;1:336–46.
22. Tang Duyu, Wei Furu, Yang Nan, Zhou Ming, Liu Ting, Qin Bing. Learning sentiment-specific word embedding for twitter sentiment classification. *ACL*. 2014;1:1555–65.
23. Weston J, Chopra S, Adams K. # tagspace: semantic embeddings from hashtags. 2014.
24. Geoffrey E, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Comput*. 2006;18(7):1527–54.
25. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. *Adv Neural Inf Process Syst*. 2007;19:153–60.
26. Li G, Zhu H, Cheng G, Thambiratnam K, Chitsaz B, Yu D, Seide F. Context-dependent deep neural networks for audio indexing of real-life data. In: *IEEE spoken language technology workshop (SLT)*. 2012. p. 143–8.
27. Brin S, Page L. Reprint of: the anatomy of a large-scale hypertextual web search engine. *Comput Netw*. 2012;56(18):3825–33.
28. Mortensen EN, Barrett WA. Interactive segmentation with intelligent scissors. *Graph Models Image Process*. 1998;60(5):349–84.
29. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst*. 2012:1097–105.
30. Hinton G, Deng L, Dong Y, Dahl GE, Mohamed Abdel-rahman, Jaitly Navdeep, Senior Andrew, Vanhoucke Vincent, Nguyen Patrick, Sainath Tara N, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process Mag*. 2012;29(6):82–97.
31. Alexandrescu A, Kirchhoff K. Factored neural language models. In: *Proceedings of the human language technology conference of the NAACL, companion, volume: short papers. Association for computational linguistics*; 2006. p. 1–4.
32. Luong T, Socher R, Manning CD. Better word representations with recursive neural networks for morphology. *Vancouver: CoNLL*; 2013. p. 104–13.
33. Lazaridou A, Marelli M, Zamparelli R, Baroni M. Compositionally derived representations of morphologically complex words in distributional semantics. *ACL*. 2013;1:1517–26.
34. Kilgariff A, Grefenstette G. Introduction to the special issue on the web as corpus. *Comput Linguis*. 2003;29(3):333–47.
35. Socher R, Pennington J, Huang EH, Ng AY, Manning CD. Semi-supervised recursive autoencoders for predicting sentiment distributions. In: *Proceedings of the conference on empirical methods in natural language processing. Association for computational linguistics*; 2011. p. 151–61.
36. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science*. 2006;313(5786):504–7.
37. Hinton GE, Zemel RS. Autoencoders, minimum description length and helmholtz free energy. *Adv Neural Inform Process Syst*. 1994:3–10.
38. Smolensky P. Information processing in dynamical systems: foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science; 1986.
39. Hinton GE. Training products of experts by minimizing contrastive divergence. *Neural Comput*. 2006;14(8):1771–800.
40. Socher R, Huval B, Manning CD, Ng AY. Semantic compositionality through recursive matrix-vector spaces. In: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. Association for computational linguistics*; 2012. p. 1201–11.
41. Socher R, Bauer J, Manning CD, Ng AY. Parsing with compositional vector grammars. *ACL*. 2013;1:455–65.
42. Socher R, Lin CC, Manning C, Ng AY. Parsing natural scenes and natural language with recursive neural networks. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011. p. 129–36.
43. Collobert R. Deep learning for efficient discriminative parsing. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011. p. 224–32.
44. Market Sentiment. <http://www.investopedia.com/>.
45. Pang B, Lee L. Opinion mining and sentiment analysis. *Found Trends Inf Retrieval*. 2008;2:1–35.
46. Loughran T, McDonald B. When is a liability not a liability? Textual analysis, dictionaries. *J Finance*. 2011;66:35–65.
47. Mao H, Gao P, Wang Y, Bollen J. Automatic construction of financial semantic orientation lexicon from large scale Chinese news corpus. In: *7th Financial risks international forum*; 2014.
48. Steinwart I, Christmann A. Support vector machine. Berlin: Springer; 2008.
49. Saif H, He Y, Alani H. Semantic sentiment analysis of Twitter. *The semantic Web-ISWC 2012*. Berlin: Springer; 2012. p. 508–24.

50. Silva N, Hruschka E, Hruschka E. Tweet sentiment analysis with classifier ensembles. *Decis Support Syst.* 2014;66:170–9.
51. Fersini E, Messina E, Pozzi FA. Automatic construction of financial semantic orientation lexicon from large scale Chinese news corpus. *Decis Support Syst.* 2014;68:26–38.
52. Potts C, Pearson K. From frequency to meaning: vector space models of semantics. *J Artif Intell Res.* 2010;37:141–88.
53. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Inf Process Manag.* 1988;24(5):513–23.
54. Robertson SE, Walker S. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval*. New York: Springer Inc.; 1994. p. 232–41.
55. Bengio Y, et al. Learning deep architectures for AI. *Found Trends Mach Learn.* 2009;2(1):1–127.
56. Chen H, Chiang RHL, Storey VC. Business intelligence and analytics: from big data to big impact. *MIS Quart.* 2012;36:4.
57. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: *IEEE computer society conference on computer vision and pattern recognition, CVPR 2005*, vol. 1. 2005. p. 886–93.
58. Lowe DG. Object recognition from local scale-invariant features. In: *The proceedings of the seventh IEEE international conference on computer vision*, vol. 2. 1999. p. 1150–7.
59. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. [arXiv:1409.0473](https://arxiv.org/abs/1409.0473). 2014.
60. Coates A, Ng AY. The importance of encoding versus training with sparse coding and vector quantization. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011. p. 921–8.
61. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. [arXiv:1207.0580](https://arxiv.org/abs/1207.0580). 2012.
62. Fan J, Han F, Liu H. Challenges of big data analysis. *Natl Sci Rev.* 2014;1(2):293–314.
63. Abadi M, Agarwal A, Barham P, Brevdo E. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proc Assoc Comput Linguis.* 2002;66:417–24.
64. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: *Proceedings of the conference on empirical methods in natural language processing*, vol. 66; 2002. p. 79–86.
65. Kiritchenko S, Zhu X, Mohammad S. Sentiment analysis of short informal texts. *J Artif Intell Res.* 2014;50:723–62.
66. Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology.* 1982;143:29–36.
67. Bermingham ML, Pong-Wong R, Spiliopoulou A, Hayward C, Rudan I, Campbell H, Wright AF, Wilson JF, Agakov F, Navarro P, Haley CS. Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Sci Rep.* 2015;5:10312.
68. Torgo L. *Data mining with R*. Boca Raton: CRC Press; 2010.
69. Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. *Bioinformatics.* 2007;23(19):2507–17.
70. Pearson K. X. On the criterion that a given system of deviations from the probable in the case of a corysystem of variables is such that it can be reasonably supposed to have arisen from random sampling. *Lond Edinburgh Dublin Philos Mag J Sci.* 1900;50:157–75.
71. Fisher R. Dispersion on a sphere. *Proc R Soc Lond.* 1953;217:295–305.
72. Grönaauer A, Vincze M. Using dimension reduction to improve the classification of high-dimensional data. [arXiv:1505.06907](https://arxiv.org/abs/1505.06907). 2015.
73. Le Q, Mikolov T. Distributed representations of sentences and documents. In: *International conference on machine learning*, vol. 31. 2014.
74. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: *Workshop at ICLR*. 2013.
75. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: *Workshop at ICLR*. 2013.
76. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997;9:1735–80.
77. Chellapilla K, Puri S, Simard P. High performance convolutional neural networks for document processing. In: *Tenth international workshop on frontiers in handwriting recognition*. Seattle: Suvisoft; 2006.
78. Goodfellow I, Bengio Y, Courville A. *Deep learning*. MIT Press; 2016. <http://www.deeplearningbook.org>.
79. Bengio Y, LeCun Y, et al. Scaling learning algorithms towards AI. *Large Scale Kernel Mach.* 2007;34(5):1–41.
80. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell.* 2013;35(8):1798–828.
81. Bengio Y. Deep learning of representations: looking forward. In: *International conference on statistical language and speech processing*. Berlin: Springer; 2013. p. 1–37.
82. Calandra R, Raiko T, Deisenroth MP, Pouzols FM. Learning deep belief networks from non-stationary streams. In: *International conference on artificial neural networks*. Berlin: Springer; 2012. p. 379–86.
83. Zhou G, Sohn K, Lee H. Online incremental feature learning with denoising autoencoders. In: *Artificial intelligence and statistics*. 2012. p. 1453–61.
84. Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, Senior A, Tucker P, Yang K, Le QV, et al. Large scale distributed deep networks. In: *Advances in neural information processing systems*. 2012. p. 1223–31.
85. Coates A, Huval B, Wang T, Wu D, Catanzaro B, Ng A. Deep learning with cots hpc systems. In: *International conference on machine learning*. 2013. p. 1337–45.
86. Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: a deep learning approach. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*; 2011. p. 513–20.
87. Chopra S, Balakrishnan S, Gopalan R. Dlid: deep learning for domain adaptation by interpolating between domains. In: *ICML workshop on challenges in representation learning*, vol. 2; 2013.

88. Larochelle H, Bengio Y, Louradour J, Lamblin P. Exploring strategies for training deep neural networks. *J Mach Learn Res*. 2009;10:1–40.
89. Olshausen AB, Field DJ. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Res*. 1997;37(23):3311–25.
90. Hinton G, Salakhutdinov R. Discovering binary codes for documents by learning deep generative models. *Topics Cogn Sci*. 2011;3(1):74–91.
91. Salakhutdinov R, Hinton G. Semantic hashing. *Int J Approx Reas*. 2009;50(7):969–78.
92. Le QV. Building high-level features using large scale unsupervised learning. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP). 2013. p. 8595–8.
93. Vincent P, Larochelle H, Bengio Y, Manzagol PA. Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th international conference on Machine learning*. New York: ACM; 2008. p. 1096–103.
94. Ranzato M, Szummer M. Semi-supervised learning of compact document representations with deep networks. In: *Proceedings of the 25th international conference on machine learning*. New York: ACM; 2008. p. 792–9.
95. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 2013.
96. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 2013.
97. Kim Y. Convolutional neural networks for sentence classification. In: *Proceedings of EMNLP*. 2014.
98. Le QV, Zou WY, Yeung SY, Ng AY. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR). 2011. p. 3361–8.
99. Gers F, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. *Neural Comput*. 2000;12:2451–71.
100. Graves A. *Supervised sequence labelling with recurrent neural networks*. Heidelberg: Springer; 2012.
101. Bastien F, Lamblin P, Pascanu R, Bengio Y. Theano: new features and speed improvements. In: *NIPS workshop on deep learning and unsupervised feature learning*. 2012.
102. Bergstra J, Breuleux O, Bastien F, Bengio Y. Theano: a CPU and GPU math expression compiler. In: *Python for scientific computing conference*. 2012.
103. Bergstra J, Breuleux O, Bastien F, Lamblin P. Thumbs up? Sentiment classification using machine learning techniques. *Python in science*, vol. 9. 2015.
104. Pang B, Lee L, Vaithyanathan S. TensorFlow: large-scale machine learning on heterogeneous distributed systems. In: *Preliminary white paper*, vol. 9. 2015.
105. Blitzer J, Dredze M, Pereira F, et al. Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. *ACL*. 2007;7:440–7.
106. Wang S, Manning CD. Baselines and bigrams: simple, good sentiment and topic classification. In: *Proceedings of the 50th annual meeting of the association for computational linguistics: short papers*, vol. 2. Association for computational linguistics; 2012. p. 90–4.
107. Tan C-M, Wang Y-F, Lee C-D. The use of bigrams to enhance text categorization. *Inform Process Manag*. 2002;38(4):529–46.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---