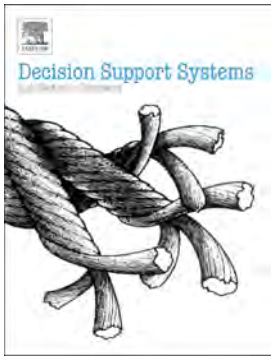


Deep neural networks understand investors better

Nader Mahmoudi, Paul Docherty, Pablo Moscato

PII: S0167-9236(18)30099-X
DOI: doi:10.1016/j.dss.2018.06.002
Reference: DECSUP 12960

To appear in: *Decision Support Systems*

Received date: 25 January 2018
Revised date: 15 May 2018
Accepted date: 13 June 2018

Please cite this article as: Nader Mahmoudi, Paul Docherty, Pablo Moscato , Deep neural networks understand investors better. Decsup (2018), doi:10.1016/j.dss.2018.06.002

Deep Neural Networks Understand Investors Better

Nader Mahmoudi

Department of Finance, Newcastle Business School, The University of Newcastle

Paul Docherty

Department of Banking and Finance, Monash Business School, Monash University

Pablo Moscato

Department of Computer Science and Software Engineering, School of Electrical Engineering and Computing, The University of Newcastle

Abstract

Studies that seek to examine the impact of sentiment in financial markets have been affected by inaccurate sentiment measurement and the use of inappropriate data. This study applies state-of-the-art techniques from the domain-general sentiment analysis literature to construct a more accurate decision support system that generates demonstrable improvement in investor sentiment classification performance compared with previous studies. The inclusion of emojis is shown significantly improve sentiment classification in traditional algorithms. Moreover, deep neural networks with domain-specific word embeddings outperform the traditional approaches for the classification of investor sentiment. The approach to sentiment classification outlined in this paper can be applied in future empirical tests that examine the impact of investor sentiment on financial markets.

Keywords: Investor Sentiment, Domain-specific, Emojis, Deep Neural Network (DNN), Word Embeddings, StockTwits

Email addresses: Nader.Mahmoudi@uon.edu.au (Nader Mahmoudi), Paul.Docherty@monash.edu (Paul Docherty), Pablo.Moscato@newcastle.edu.au (Pablo Moscato)

1. Introduction

Although the neoclassical finance paradigm of efficient markets provides the proposition that stock returns are unpredictable (Fama, 1970), a large body of contradictory empirical evidence has brought this theory into question (Baker and Wurgler, 2000; Cochrane, 2000). In light of this evidence, *behavioral finance* has been proposed as an alternative theoretical paradigm to explain stock returns. The key implication of behavioral finance is that the emotions and moods of investors play an important role in financial decisions (Nofsinger, 2005). Moreover, the presence of irrationality and the emotive basis of decisions made by noise-traders, who comprise a relatively large proportion of stock market participants (Black, 1986), has resulted in investor sentiment being considered to influence investor decision-making, and hence stock returns. This new paradigm of stock market behavior has resulted in the need to develop an accurate measure of investor sentiment (Chan and Chong, 2017).

Despite a large number of studies proposing a relationship between investor sentiment extracted from social media and stock market returns, there is no consensus in empirical studies whether this theoretical relationship is supported in the data. Proponents of behavioral finance argue that this lack of empirical evidence can be attributed mainly to problems with the measurement of investor sentiment through social networks in existing studies of financial markets. These problems include: the absence of an accurate approach for measuring investor sentiment (Renault, 2017; Oh and Sheng, 2011); use of datasets from platforms that do not accurately represent investors (Bollen et al., 2011; Ranco et al., 2015); and the use of short sample periods (Bollen et al., 2011; Li et al., 2018). Motivated by these problems, this study applies recent advances in the domain-general sentiment analysis literature to data from a finance-related social media platform to construct a more accurate decision support system in the context of investor sentiment classification.

The collection of investor sentiment data from Internet-based microblogs overcomes issues that have been identified from the use of questionnaires, such as errors due to impaired questionnaire design (Brace, 2008) and inaccurate or untruthful participant responses (Singer, 2002). While previous studies have sought to measure investor sentiment using other mi-

croblogs, such as Twitter (Bollen et al., 2011; Ranco et al., 2015), StockTwits should provide a more relevant source of information to measure investor sentiment, given its focus on stock-related information (Oliveira et al., 2016). However, it is difficult to use basic classification approaches to classify investor sentiment in StockTwits given the distinct properties of the texts in this microblog. First, the terminology in StockTwits employs everyday English words but in ways that carry specific financial and investment meanings (Oliveira et al., 2016). Second, StockTwits is also characterized by the use of non-text characters to convey feelings and beliefs, such as emojis and emoticons (Novak et al., 2015). Moreover, the posts made by users (investors) comprise a more prominent use of negation, sarcasm, and domain-specific analogies that are very hard to extract by hand-crafted features (Shirani-Mehr, 2014).

Recent developments in domain-general sentiment classification may provide insights that can be used to improve the classification of investor sentiment from social media data. For instance, non-text features such as emojis has been shown to improve sentiment classification (Novak et al., 2015), a range of domain-general and domain-specific sentiment lexicon resources have been constructed for sentiment classification (Baccianella et al., 2010; Deng et al., 2017; Oliveira et al., 2016), various word embeddings have been proposed for feature engineering in natural language processing (NLP) (Mikolov et al., 2013a; Pennington et al., 2014), and deep neural networks have been adopted over several NLP tasks (Kalchbrenner et al., 2014). However it is important to use approaches to sentiment classification that is specific to the finance context, given domain independent lexicons or general word embeddings are likely to perform well in the financial domain (Li and Shah, 2017; Oliveira et al., 2016). Despite the above advancements in NLP, extant studies that classify investor sentiment have only applied simple structures that rely on hand-crafted feature types and shallow classification techniques. Through the way of developing a state-of-the-art classification technique, this study examines the incorporation of non-text features (emojis), the development of domain-specific word embeddings, and the use of deep learning to classify investor sentiment.

Deep neural networks (DNNs) reach state-of-the-art performance in most of the NLP problems without any need for enhanced pre-engineered features (Kalchbrenner et al., 2014). These models can capture deep local features by convolution kernels or capture long-distance

dependencies by memory units over the input sentences (Wang et al., 2016). The success of word embedding construction algorithms, which take a large corpus as input and produce a high-dimensional vector space (Mikolov et al., 2013a; Pennington et al., 2014), has led to an increase in the implementation of DNNs on NLP problems. The word embeddings play a strongly significant role in solving the NLP problems as they succeed in representing semantic and syntactic relationships between words in a context. Meanwhile, various intrinsic or extrinsic methods have been proposed by researchers in order to evaluate the word embeddings: similarity (relatedness), analogy, POS tagging, and sentiment classification (Schnabel et al., 2015). Having developed such dynamic research in the domain-general NLP, the existent studies that examine the investor sentiment have pursued simple approaches with basic feature types and shallow classification techniques. To date, DNNs have not been implemented for the classification of investor sentiment.

The contributions of this study are three-fold, leading to a more accurate decision support system to facilitate sentiment classification in financial markets. First, the inclusion of non-text features, namely emojis, is shown to improve investor sentiment classification. Second, this study evaluates GloVe and Word2Vec to analyze their ability in capturing domain-specific word similarities compared with domain-general word embeddings. This is carried out through a novel domain-specific evaluation method called the FinSim Index, which represents the similarity between two words in the finance context. Finally, different types of deep neural networks are constructed for the problem of investor sentiment and shown to further improve sentiment classification compared with traditional classifiers. By means of a qualitative analysis, it is demonstrated that the deep neural networks detect abstract-level feature types such as sarcasm and irony, which potentially explains their superiority.

This paper continues with a literature review in section 2 and a discussion of the methodology in section 3. The results are reported in section 4, including a discussion of emojis, word embeddings, and deep neural networks, while section 5 gives the conclusion, describes the limitations, and foreshadows future work.

2. Related Literature and Research Questions

News websites, social networks, and weblogs provide modern investors with the opportunity to exchange information and opinions about financial markets with high frequency (Sun et al., 2014). Since the advent of the Internet, various techniques have been utilized by researchers in order to use this information to extract measures of investor sentiment. These methods can be classified into two main groups: lexicon-based techniques and machine learning techniques. The use of these lexicon-based approaches in the financial domain was initiated by Tetlock et al. (2008), who constructed a daily measure of the sentiment using daily content from a popular Wall Street Journal (WSJ) column. This measure is called the *pessimism factor* since it is highly related to words with negative polarity.

A key limitation with the pessimism factor is that it is constructed by categorizing words according to the General Inquirer's Harvard IV-4 dictionary. Such dictionaries may be limited in their ability to assign sentiment to words used in the financial context, due to the use of domain-specific language. In order to overcome this drawback, Dougal et al. (2012) and García (2013) have constructed an alternative investor sentiment measure by using a domain-specific dictionary that is developed from a large sample of 10k financial reports (Loughran and McDonald, 2011). In another in depth research, Oliveira et al. (2016) have created a domain-specific lexicon using data from StockTwits, a social media platform designed specifically for the purpose of sharing perspectives of stock market investing. The development of this lexicon is important given language is expressed using a range of unique features on social media platforms, however one limitation is that lexicon-based sentiment classification approaches are not able to capture a range of linguistic structures that have become pervasive amongst social media users, including the use of emojis and emoticons, Internet slang, acronyms and sarcasm.

The vast amount of investment sentiment-related data that is publicly available on the Internet has resulted in tremendous growth in the use of machine learning-based approaches for investor sentiment classification. Naïve Bayes, Decision Trees, and Support Vector Machines are commonly used for classifying texts from stock message board postings on Yahoo Finance,

Twitter, and StockTwits (Antweiler and Frank, 2004; Al Nasseri et al., 2014; See-To and Yang, 2017; Li et al., 2018). To boost these approaches, different feature selection and extraction techniques including bag-of-words (BoW), TF-IDF weighting scheme, and , information gain criteria are applied. For example, Wang et al. (2017) identify that superior performance in StockTwits sentiment classification is achieved by using uni-grams as the features and Support Vector Machine as the classifier. Given each of these previous studies have been undertaken by feeding various types of manually-crafted feature sets to the machine learning classifiers, none of these studies have considered emojis in the classification problem of investor sentiment. This study analyzes the emojis' discriminatory power, despite their domain-specific pattern of usage, to respond following research question:

Research Question 1: Will the classification model including emojis outperform that without emojis?

The use of word embeddings has been demonstrated to improve sentiment classification across general domain settings, although this approach has had limited application in the context of social media investor sentiment classification. Word embeddings, which represent the distributional semantics of words in a context, have recently experienced a growing interest among researchers in NLP. Each word is transformed into a d-dimensional embedding vector of real numbers, capturing semantic and syntactic similarities between words. Two widely used algorithms, Word2Vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014), take a large corpus as an input and produce these high-dimensional vectors working as unsupervised learning algorithms. GloVe examines the co-occurrence matrix of the words in constructing the word embeddings, whereas Word2Vec trains a simple neural network with one hidden layer. These vectors are used efficiently as features in a variety of applications, including information retrieval, document classification, question answering, named entity recognition, and parsing (Pennington et al., 2014). However, these domain-general word embeddings are not able to perfectly capture domain-specific similarities, especially in StockTwits where investors have constructed their own language. In order to overcome this domain-specificity, Li and Shah (2017) have trained domain-specific word embedding over a dataset from StockTwits with the aim of building a finance sentiment lexicon and reported that the domain-specific word

embeddings result in better sentiment lexicons than the domain-general word2vec embeddings. We extend on that study by constructing domain-specific word embeddings on a significantly bigger corpus using both GloVe and Word2Vec. Moreover, while Li and Shah (2017) only use extrinsic evaluation scheme, we aim to evaluate the quality of these word embeddings in capturing finance context similarities intrinsically, using our novel 'FinSim index', and extrinsically, feeding them into various DNNs. Thus, our second research question is as follows:

Research Question 2: To what extent do the domain-specific word embeddings outperform the domain-general ones, both intrinsically and extrinsically?

Deep learning is one of the popular machine learning techniques that has commanded attention in various complex artificial intelligence problems including computer vision (Krizhevsky et al., 2017), speech recognition (Hinton et al., 2012), and machine translation (Luong et al., 2015). After the remarkable advancements in the construction of word embeddings, the demand has arisen for highly-sophisticated learning models that can effectively extract higher level features from these vectors. With different architectures, DNNs have been adapted successfully to natural language processing problems, specifically sentiment analysis. For the first time, Kalchbrenner et al. (2014) implemented Convolutional Neural Network (CNN) on various sentence modeling problems and classify sentiment across a Twitter dataset, resulting in a 25% reduction in error compared with the state-of-the-art traditional classification systems. Furthermore, different versions of Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014), have also been tested on various NLP tasks including sentiment classification (Yin et al., 2017). These models can capture long-term semantic and syntactic dependency of the texts, whereas CNNs focus on the local features through the convolutional and pooling layers embedded inside them. Wang et al. (2016) have introduced joint CNN and RNN architecture to combine their advantageous characteristics of simultaneously extracting local and long-term features respectively. The word embeddings trained by Word2Vec/GloVe have been commonly used for various NLP tasks as they tend to lead DNNs in solving the semantic and syntactic sparsity (Wang et al., 2016). In an intuitive study, Kim (2014) takes

advantage of pre-trained word embeddings in training a simple CNN model for several sentence classification datasets. He shows that this model outperforms the model with random embeddings initialization. This study comprehensively evaluates the performance of DNNs in the problem of investor sentiment, having a response to the following research question. Moreover, this enables us to extrinsically evaluate the quality of word embeddings through the classification performance of DNNs.

Research Question 3: Do DNNs outperform traditional classifiers in the problem of investor sentiment?

Answering these three major research questions, this study yields an inclusive analysis regarding the role of a modern feature type, emojis, construction and evaluation of domain-specific word embeddings, and comprehensively comparison of various DNN models with state-of-the-art traditional classifiers. The key contribution of this study is the application of these approaches in the financial context, which is particularly important given the identified need for improved measures of investor sentiment classification (Renault, 2017) and the poor performance of domain-general lexicons and word embeddings in the financial domain (Li and Shah, 2017; Oliveira et al., 2016).

3. Materials and Methods

In this section, we first discuss the characteristics of the data from StockTwits together with the experimental setup. Then, the traditional classification paradigm is described briefly, where the best-performing classifier will help us to answer research question 1. Later, two main algorithms for the development of word-embeddings are discussed shortly in subsection 3.3. Moreover, a new intrinsic approach has been designed to evaluate the word-embeddings, presenting an answer for research question 2. Finally, subsection 3.4 reviews various deep neural networks structures, where we aim to respond research question 3.

3.1. Data from StockTwits and Experimental Setup

StockTwits is a social media platform designed for investors wherein they share ideas, beliefs, and/or feelings about financial markets behavior. It is a place for users to observe

traders and investors, produce posts and contribute to conversations related to the market and individual stocks. Here, amateur investors can meet and interact with professionals freely. The streams in StockTwits contain ideas, links, charts, and financial data expressed within 140 characters. By the end of 2016, more than 63 million messages had been posted by 250,000 users.

[Table 1 about here.]

Table 1 provides a brief statistics of collected messages from StockTwits between June 2008 and December 2016. As shown in Table 1, users posted 63,647,533 messages in total between June 2008 and December 2016, which include 82.20% unlabeled messages, 14.39% positive messages, and 3.41% negative messages. For the purpose of this research, all unlabeled messages have been fed into two word embedding generation algorithms, GloVe and Word2Vec, to map the words into the high-dimensional space of embeddings. Furthermore, a random sample of five percent of both the positively and negatively labeled messages has also been compiled. This sample, which should be representative of the population of labled messages while allowing for computational efficiency, comprises 458,067 bullish and 108,659 bearish messages[1]. Following the approach to sentiment classification used by the StockTwits platform, we adopt a binary classification of sentiment, bullish versus bearish. All classifiers and algorithms have been trained and tested on this dataset in order to have a consistent comparison. All messages have been put through some general pre-processing tasks, including the replacement of URLs with <url>, cashtags with <cashtag>, hashtags with <hashtag>, user mentions with <usertag>, and real numbers with <number>, collapsing letter repetitions (e.g. "haaaaapppppppy" and "Coooool" will become "haaapppy" and "Coool", respectively), expanding contractions (e.g. "I've" will be replaced with "I have"), and discarding tokens with occurrences less than five.

Due to the unbalanced nature of messages from StockTwits (with approximately one bearish message to four bullish messages), we have implemented the popular heuristic of

[1]As a robustness test, analysis was also undertaken using a balanced dataset of 434,636 messages with both positive and negative sentiment and the results were robust to this alternative approach to sampling.

class-dependent misclassification costs to bias the classifiers towards the minority class, where the misclassification cost of the minority class equals to imbalance ratio of the training dataset (Weiss, 2004). Moreover, to properly validate the classification models against the dynamics of messages through time, we have adopted the rolling window scheme (Moro et al., 2014) with 20 equally-sized windows ordered by time. For every window, we have measured the Matthew correlation coefficient (MCC), given this approach has been reported to be the most appropriate measure of performance in unbalanced datasets (Boughorbel et al., 2017). MCC, measured using Equation 1, is regarded as a realistic measure which thoroughly describes the confusion matrix with a single number between $[-1, 1]$[2]. In addition, we have used the Wilcoxon Sum-Rank Test (WSuRT) (Vidakovic, 2013) to statistically confirm the performance of the classifiers and validate the conclusions under the assumption that each window is an independent and random sample.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (1)$$

3.2. The Traditional Machine Learning Paradigm

Various feature types have been examined thoroughly in traditional sentiment classification problems (Pang and Lee, 2008), including *n*-grams, negation, and emoticons/emojis. In this study, we have followed Pang et al. (2002) to examine the effect of unigrams, bi-grams, tri-grams, and negation tag in the problem of investor sentiment. Undoubtedly, emojis play a decisive role in determining the sentiment polarity of informal and short texts in social media, weblogs, or comments. As discussed before, however, the language used by people in StockTwits differs significantly from other social networks. For example, the word “red" carries a pessimistic meaning in StockTwits while it would be interpreted simply as a color in other social networks. Emojis also have different usage patterns in StockTwits. 🚀 (rocket), 💰 (money bag), 🐻 (bear face), 💩 (pile of poo), 🐂 (ox), 📈 (chart increasing), and 📉 (chart decreasing) are some of the emojis commonly used by investors on the StockTwits platform

[2] A Matthew correlation coefficient of +1 shows the ideal prediction, 0 no better than the random prediction, and -1 represents the absolute disagreement between predicted and actual label.

in order to express their feelings and ideas. For the first time in the field of investor sentiment classification, we explore the effect of emojis on financial text labeling.

Three relatively popular and high-performing classification algorithms in the problem of investor sentiment, including Naïve Bayes (NB) (Manning et al., 2008), Maximum Entropy (MaxEnt) (Berger et al., 1996), and Support Vector Machine (SVM) (Scholkopf and Smola, 2001), will be implemented to identify the most successful one in the problem of investor sentiment classification. NB is a simple probabilistic classifier which is based on Bayesian Theorem and assumes independence among features of the observations. MaxEnt is another probabilistic classifier that estimates the probabilities of possible outcomes of a dependent variable using a set of independent features. Whereas, SVM, which is a larger marginal classifier rather than a probabilistic classifier, separates the observations in different classes, optimally keeping the margin as large as possible. The best traditional algorithm chosen here will play a baseline role for the rest of analysis undertaken in this study.

3.3. Word Embeddings

The word vectors, also called word embeddings, capture semantic and syntactic characteristics of words over the corpus. Thus, semantically and syntactically similar words will be mapped to nearby points. Word embeddings, which are input for the deep neural networks as well, are constructed by two well-known algorithms, GloVe and Word2Vec. Applying these two algorithms, we have also built new word embeddings using unlabeled messages collected from StockTwits in order to evaluate their performance in capturing domain-specific similarities in finance.

3.3.1. Skip-gram with Negative Sampling (SGNS)

Briefly, SGNS (Mikolov et al., 2013b) is a predictive approach that tries to find context words surrounding a given target word. Using a fully connected neural network with a single hidden layer, it aims to maximize the average of the sum of log probabilities through the following objective function:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T}\sum_{-n\leq j\leq n, j\neq 0} \log p(w_{t+j}|w_t) \tag{2}$$

where T is corpus size, n is context size, and $p(w_{t+j}|w_t)$ is calculated by the following softmax function:

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^\top v_{w_I})}{\sum_{w \in W} \exp({v'_w}^\top v_{w_I})} \tag{3}$$

where W is vocabulary size and v_w and v'_w are "input" and "output" embedding vector of word w. However, this will lead to a very high computation cost of $\nabla p(w_O|w_I)$ due to the size of W, which is usually large, and, therefore, two options have been introduced in order to make it computationally efficient (Mikolov et al., 2013b).

First, the **sub-sampling** scheme has been proposed to deal with frequent words such as "in", "the", and "a", as they usually provide less information than rare words. Second, the **negative sampling** has been presented based on the skip-gram model but with a different objective function to approximate the loss of softmax with the aim of reducing computation time.

3.3.2. Global Vectors (GloVe)

On the other hand, GloVe (Pennington et al., 2014) forms the co-occurrence matrix X each of whose elements, X_{ij}, represents the number of times word j appears in the context of word i. The word context is defined by a variable window size. During construction of the co-occurrence matrix, the decreasing weighting function of $1/d$ applies for the pairs that appear d words away from the center word as they may carry less relevant information.

The soft constraint for each word pair is defined as follows:

$$w_i^T w_j + b_i + b_j = \log(X_{ij}) \tag{4}$$

where w_i is vector for center word and w_j is vector for the context word and b_i and b_j are their scalar biases, respectively.

In the end, the cost function below, a weighted least squares regression model, will be minimized:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log(X_{ij}))^2 \tag{5}$$

where V is the size of the vocabulary and f is weighting function designed to reduce the effect

of extremely common word pairs. The authors have chosen the following function:

$$f(X_{ij}) = \begin{cases} (\frac{x}{x_{max}})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{o.w.} \end{cases} \tag{6}$$

where $x_{max} = 100$ and $\alpha = 0.75$, suggested in the corresponding paper (Pennington et al., 2014).

3.3.3. Word Embeddings Evaluation

[Table 2 about here.]

Two main schemes have been introduced to evaluate the word embeddings: extrinsic and intrinsic. Extrinsic evaluation methods use word embeddings as an input for another task such as named-entity recognition, part-of-speech tagging, or sentiment classification (Pennington et al., 2014) with their particular performance measure. However, intrinsic methods assess the quality of word embeddings by evaluating syntactic and semantic relationships between words by use of a set of pre-selected query terms (Schnabel et al., 2015). Similarity (or relatedness) is an example of intrinsic approaches where the aim is to measure the correlation between the similarity scores of query terms and cosine similarity as computed by the word embeddings. However, existing query datasets are not suitable to evaluate finance word embeddings as they do not cover any domain-specific query terms. Therefore, we have constructed a dataset of 158 query terms for finance, called *FinSim*, to assess the word embeddings intrinsically. Table 2 shows a few examples of queries using the FinSim Index and cosine similarities calculated from GloVeST and Word2VecST (domain-specific word embeddings) and GloVe[3] and Word2Vec[4] (domain-general word embeddings). The FinSim index represents the similarity of words in the finance context, independently scored by five finance experts[5], and scales between $[-1, 1]$. Then, the Pearson correlation between this

[3]Available on `http://nlp.stanford.edu/data/glove.6B.zip`

[4]Available on `https://code.google.com/archive/p/word2vec/`

[5]The experts are all tenured academics located within the finance department at an Australian university.

FinSim index and cosine similarities reveals the quality of word embeddings and their ability to represent finance-related syntactic and semantic relationships.

3.4. Deep Learning Paradigm

"Deep learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level" (LeCun et al., 2015, p. 1). Deep neural networks automatically capture the representation of words, conferred contextual information, from a raw input corpus for a particular task, independent of any hand-crafted features. While the traditional limitation of deep neural networks is that they require substantially more computation time compared with traditional classifiers, therefore limiting their efficacy in the field, the development of graphics processing units (GPUs) has substantially reduced computation time to enable these techniques to be applied more widely. We will discuss word embeddings as well as each of these architectures in the following sub-sections.

3.4.1. The Convolutional Neural Network (CNN)

The CNN implemented for investor sentiment classification in this study, following Kim's study Kim (2014), consists of four primary layers. The corresponding CNN is constructed by an input layer, convolution layer, max-pooling layer, and fully connected layer (see Figure 1). Below, we have discussed each layer with the relative mathematical formulation.

[Figure 1 about here.]

Input Layer treats the input sentence (tweet on StockTwits in our case) as a sequence of n words, each of which is represented by a d-dimensional vector of embedding: $[x_1, x_2, x_3, ..., x_n]$ where $x_i \in \mathbb{R}^d \; \forall \; i = 1, 2, 3, ..., n$. These embedding vectors are either initialized randomly or fed from pre-trained word embeddings constructed by GloVe or Word2Vec.

Convolution Layer aims to capture local features that concurrently appear in the previous layer by a set of learnable filters called convolution kernels. Mathematically, the weight matrix for the convolution filter is $w \in \mathbb{R}^{h \times d}$, which will be applied to the window of h words

with an embedding dimension of d. After convolving every possible window of words, the feature map c becomes:

$$c = [c_1, c_2, c_3, ..., c_{n-h+1}] \tag{7}$$

where $c \in \mathbb{R}^{n-h+1}$ and the convolution filter c_i for position i in the sentence is calculated by:

$$c_i = f(\mathbf{w} \cdot x_{i:i+h-1} + b) \tag{8}$$

where $b \in \mathbb{R}$ is the bias and f is a non-linear activation function.

Max-pooling Layer addresses the most important features by pooling over every feature map bearing a close resemblance to the process of feature selection in natural language processing. Thus, the pooled feature map, p, will be calculated by:

$$p = [\max(c_1, c_2, c_3, ..., c_{n-h+1})] \tag{9}$$

Finally, the concatenated and flattened pooled feature maps are passed through a high dimensional dense layer - known as ***the fully connected layer*** and fed into the output layer whose output is the class probabilities. The output layer computes these probabilities by soft-max activation as follows:

$$P(y = j|\mathbf{x}, \mathbf{w}, b) = \text{softmax}_j(\mathbf{x}^T\mathbf{w} + b) = \frac{e^{\mathbf{x}^T\mathbf{w}_j + b_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^T\mathbf{w}_k + b_k}} \tag{10}$$

where w_k and b_k are the weight vector and bias of the k-th class.

3.4.2. The Recurrent Neural Network (RNN)

The recurrent neural network (RNN), as an extension of feed-forward neural networks, can handle variable-length sequences, having a recurrent hidden state whose activation on the current time-step is dependent on what it has seen on the earlier time step (see Figure 2(a)). Despite excellent performance on various problems such as speech recognition, language modeling, and image captioning, the original RNN is not practically able to learn long-term dependencies in the sequences Bengio et al. (1994). Two recent versions of RNNs have been proposed: Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (Cho et al., 2014). The input and output layers are the same as in CNN, so

we skip re-explaining them here. The following subsections will discuss the LSTM and GRU units (shown in Figure 2(b) and Figure 2(c) respectively).

[Figure 2 about here.]

Long Short-Term Memory (LSTM).

Incorporating the cell state C_t at time step t, the LSTM unit controls the flow of information from the previous time step. This enables it to store relevant information from early time steps and carry it over long time steps to employ in later time steps. This process takes place through three gates; the forget gate, the input gate, and the output gate (see Figuer 4(b)). The parameters are updated through the following equations:

$$f_t = \sigma(W_f \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + b_f) \tag{11}$$

$$i_t = \sigma(W_i \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + b_i) \tag{12}$$

$$\widehat{C}_t = \tanh(W_C \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}]) \tag{13}$$

$$C_t = f_t * C_{t-1} + i_t * \widehat{C}_t \tag{14}$$

$$o_t = \sigma(W_o \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + b_o) \tag{15}$$

$$h_t = o_t * \tanh(C_t) \tag{16}$$

Gated Recurrent Unit (GRU).

Like the LSTM unit, the GRU is designed to capture long-term dependencies of the input sequences but without carrying the cell state from one time step to the next. Moreover, it merges the input gate and forget gate into a single update gate that controls the degree to which past information should matter in the current time step. This is determined by:

$$Z_t = \sigma(W_z \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}]) \tag{17}$$

$$r_t = \sigma(W_r \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}]) \tag{18}$$

$$\widehat{h}_t = \tanh(W \cdot [r_t * \mathbf{h}_{t-1}, \mathbf{x}_t]) \tag{19}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widehat{h}_t \tag{20}$$

4. Results and Discussions

In this section we discuss the performance of both traditional classifiers that use various feature types and deep neural networks for classifying investor sentiment.

4.1. Traditional Investor Sentiment Classification

[Table 3 about here.]

Table 3 briefly illustrates the experimental setups for traditional classification approaches including feature extraction and classifier development. The raw messages are transformed to binary feature vectors whose element i is set after pre-processing to one if feature f_i exists in the corresponding message and zero otherwise. By removing infrequent and useless features such as misspellings, we have discarded the features that appear in less than five messages to reduce the sparsity of the input. Instead of using built-in stop-words, we have eliminated the features that appear in over 75% of the messages to remove less informative but highly frequent features. During the classifier development process, we have implemented SVM with linear kernel, MaxEnt with liblinear solver, and Multinomial NB classifier[6].

4.1.1. Which Feature Type, Which Algorithm

[Figure 3 about here.]

In our method of constructing the baseline to evaluate the effect of emojis and deep neural networks in the financial context, four different feature types with three classification techniques have been incorporated. Figure 3 shows the performance of the classifiers, Naïve Bayes (NB), Maximum Entropy (MaxEnt), and Support Vector Machine (SVM), incorporating four different feature sets, 1-grams, 1-grams with negation tag, (1,2)-grams, and (1,2,3)-grams. MaxEnt out-performs SVM on every feature type (WSuRT with p-value$\leq$0.02831). Although NB

[5]These values are chosen through a random search parameter tuning scheme.

[6]We have used Scikit-Learn API steps of developing the traditional classifier.

out-performs the SVM and MaxEnt over 1-grams and 1-grams with negation tag, it under-performs significantly MaxEnt when bi-grams and tri-grams are added to feature vectors (WSuRT with p-value$\leq$0.00011). Moreover, it can be seen that the negation tag does not lead to significant improvement in the performance of classifiers. This result implies that more sophisticated feature engineering mechanisms are required to capture such a complex linguistic structure, whereas, combining bi-grams or tri-grams with uni-grams boosts the performance regardless of the classification algorithm (WSuRT with p-value$\leq$0.01548). Moreover, the domain-specific lexicon developed by Oliveira et al. (2016) significantly under-performs the traditional classifiers in our dataset despite its high computational efficiency, with a MCC of 0.3081. Therefore, while the approach advocated by Oliveira et al. (2016) can substantially reduce computation time, our results demonstrate that this efficiency comes at a cost of a substantial reduction in classification accuracy compared with more complex models.

4.1.2. Emojis and Investors

[Figure 4 about here.]

[Figure 5 about here.]

Emojis are dynamic and dominant entities of financial social networks, in our case StockTwits, where investors actively express their feelings and opinions. This website has provided emojis for the use of investors since mid-2015. Therefore, the limited number of messages, approximately 0.8% of messages, contain at least one emoji. Emojis are now becoming exponentially more popular as a means of expressing feelings and emotions in the financial context (see Figure 5(a)). Briefly, there are 1,658 unique emojis that have been used 1,032,352 times overall by investors in 508,097 messages. Of the sentiment-labeled messages, there are only 19,376 bearish and 144,166 bullish messages in which at least one emoji has appeared[7].

[7]Figure 4 shows the emoji cloud for common emojis in bullish and bearish messages.

Therefore, to better demonstrate and analyze emojis' discriminative power in investor sentiment classification, we have implemented MaxEnt, the best classifier from previous section, on this dataset. The expected results reveal that the emojis lead to 44-47% higher MCC[8] through manipulating the discriminative power that is somehow hidden in their usage pattern. However, the presence of other feature types, bi-grams, tri-grams, and negation, does not have significant impact on the classifier's performance in the corresponding dataset with p-value$\geq$0.1051.

4.2. Word Embeddings, Domain-specific vs. Domain-general

Table 4 presents brief information about the corpus on which they have been trained. Without padding, the corpus contains 52,313,016 unlabeled messages from StockTwits which are composed of more than 838 million tokens after pre-processing. The domain-specific word embeddings, GloVeST and Word2VecST, are trained for 50 iterations to construct a 300-dimensional embedding vector of the words that appeared more than five times, taking a window size of eight and setting the parameters to their default values. By way of comparison, GloVe and Word2Vec are trained on the general datasets, Wikipedia 2014 plus Gigaword 5 with 6 billion tokens and Google News with 100 billion tokens respectively.

[Table 4 about here.]

[Figure 6 about here.]

Figure 6 displays the correlation score between the FinSim index and the cosine similarity calculated by each pre-trained word embedding. It can be easily seen that domain-specific word embeddings out-perform the domain-general word embeddings in capturing the finance context similarities. It is also shown that the SGNS has produced more high-quality word embeddings to interpret the finance-specific language used by investors than GloVe. Stressing on high demand for domain specific word embeddings, Word2Vec performs better than GloVe among domain-general word embeddings. It is able to capture some level of finance syntactic and semantic relationships owing to its extremely large training dataset.

[8]This is confirmed by WSuRT with p-value$\leq$1.083e-5.

4.3. Deep Learning Algorithms

This section discusses the performance of deep neural networks and the three-fold effect of word embeddings: convolutional versus recurrent neural networks, domain-specific versus domain-general word embeddings, and static versus non-static word embeddings. Before discussing the results, note that deep neural networks have millions of weights other than word embeddings to fit the problem at hand. The models used in this study reflect the quality of the word embeddings while performing sentiment classification. However, the difference might not be huge, especially when a reasonable amount of data is provided for them to train. Referring to Zhang and Wallace (2017) and Reimers and Gurevych (2017), we have carried out a random search scheme in order to fine-tune the critical hyper-parameters of DNNs shown in Table 5. Setting the default values for the rest of hyper-parameters, we have tested 40 distinct hyper-parameter configurations for each CNN, GRU, and LSTM[9].

[Table 5 about here.]

4.3.1. Convolution vs. Recurrent

Figure 7 illustrates the performance of deep neural networks trained over various word embeddings with static or non-static states. CNNs and RNNs, trained with non-static domain-specific word embeddings, both outperform the best traditional approach[10], extracting a greater number of hidden sentimental and semantic overtones of messages. The CNNs handle local features at different positions using convolutional filters and handle long-range relationships using pooling operations. In contrast, RNNs try to capture long-term dependencies through memory and forget gates. As the tweets are fairly short, we expected CNN to perform as well as the GRU and LSTM. However, it has underperformed the RNNs more specifically when static word embeddings are fed into the models (WSuRT with p-value$\leq$8.18e-6). Moreover,

[9]The models have been constructed in Keras API (https://keras.io/) with Tensorflow backend and trained on single Nvidia Tesla V100-SXM2-16GB GPU.

[10]MaxEnt with (1,2)-grams plus emojis gives $MCC = 0.4313$ on the same dataset (WSuRT with p-value$\leq$2.403e-6).

we do not observe significant difference between the GRU and LSTM with non-static word embeddings. Whereas, GRU performs significantly better than LSTM trained with static W2V and W2VST word embeddings.

[Figure 7 about here.]

4.3.2. Domain-specific vs. Domain-general

As Figure 7 shows, domain-specific word embeddings lead to higher MCC than the domain-general ones even in the non-static state where the network updates them over the problem at hand. As shown in subsection 4.2, this is because words have a different pattern of usage in the finance context where investors have developed their own language. Therefore, compared with their general-domain counterparts, domain-specific word embeddings are more efficient universal feature extractors that help deep neural networks better understand financial language (WSuRT with p-value$\leq$0.0245). This result is particularly noteworthy given deep neural networks have millions of weights other than word embeddings to update during the training process.

4.3.3. Static vs. Non-static

Initial word embeddings carry any information about syntactic and semantic properties of every token in the corpus where the words with similar syntactical and semantic characteristics appear close to each other. However, they do not entail any information about the characteristics of the words for the problem at hand. Therefore, the deep neural network with non-static word embedding provides a valuable opportunity to adjust word vectors and make them more specific to the problem at issue here, investor sentiment classification. Figure 8 shows the location of the top 10 polar words in the stock market context before (static state) and after (non-static state) training the LSTM with GloVeST (the best combination). Consistent with the findings in the literature, the neural network fine-tunes the word embeddings in such a way that they become distinguishable based on their sentiment too (Kim, 2014). Thus, this provides an ideal chance for the neural networks to calibrate the word embeddings to reach their highest performance. We can observe significant out-performance of DNNs

with non-static word embeddings compared to static ones, as confirmed by WSuRT with a p-value$\leq$0.0283. Whereas, GRU does not show similar performance when trained with domain-specific word embedding, GloVeST and W2VST.

[Figure 8 about here.]

4.3.4. Qualitative Analysis

In order to better understand the performance of deep neural networks, LSTM with GloVeST, we have extracted the saliency of some of the input texts where the goal is to visualize the units that contribute most to the final classification. By computing the gradient of output category with respect to the input, the saliency score demonstrates how output value changes with respect to a small change in the input (Simonyan et al., 2013). In other words, the saliency score is the absolute value of the derivative of the loss function with respect to each dimension of all input words in the corresponding sentence (Li et al., 2015). Figure 9 illustrates the gradient concentration of all input words in 12 sentences with variable length and various types of structures. With these few examples, the aim is to show how LSTM reflects different properties such as negation, sarcasm, irony, joke, and/or emojis. For the short messages, the LSTM relies mostly on discriminative features, such as emojis or sentimental words. Not surprisingly, it is able to understand some level of jokes and sarcasm by assigning higher saliency to the relevant tokens. On the other hand, it disregards insubstantial parts with lower saliency score and accumulates key information over lengthy sentences capturing long-term discriminative dependencies. Taken as a whole, Figure 9 demonstrates that deep neural networks are able to capture abstract features in the data that can not be captured by traditional classifiers; likely explaining their improved performance in investor sentiment classification. This ability to capture such abstract features that tend to be highly prevalent in social media posts indicates a strong preference for this approach, despite the reduced computational efficiency.

[10]Principal Component Analysis (PCA) (Wold et al., 1987) is used to reduce the dimension of word embedding vectors in order to ease visualization.

[Figure 9 about here.]

5. Conclusion, Limitations, and Future Work

The development of an accurate classifier of investor sentiment is required to support decision making in financial markets. Using data from StockTwits, it is shown that MaxEnt and NB outperform SVM despite their simple classification foundation with a strong independence assumption of the features. Moreover, bi-grams and tri-grams robustly boost the classification performance of investor sentiment, capturing long-range dependencies to some extent in the tweets. Although negation is one of the key grammatical rules that inverts the meaning and polarity of a sentence in multiple ways, the implemented negation tagging mechanism does not lead to significant improvement in the performance of classifiers (see Figure 3). Besides, the domain-specific lexicon does not illustrate expected performance in our dataset, confirming its limited ability to capture complex linguistic structures and entities. As discussed in Section 4.1.2, this study reveals that emojis carry very strong discriminative power in the finance context in spite of their domain-specific pattern of usage. Thus, the existence of emojis in the financial texts has contributed substantially to classification performance.

In general, deep neural networks significantly outperform traditional methods, depending on the topology and word embeddings (see Figure 7). As we have discussed previously, LSTM and GRU unexpectedly perform better than CNN, although the StockTwits messages are quite short and therefore suitable for CNN to learn local features. LSTM demonstrates robust ability to capture long-term discriminative dependencies without any feature engineering. It is able to focus on the linguistic entities such as emojis, negation, and sarcasm to some degree. Domain-specific word embeddings produce better DNN models of investor sentiment classification and achieve comparably higher MCC. The domain-specific word embeddings have presented a consistent performance in capturing finance-context similarities compared with general-domain word embeddings, as shown by the intrinsic evaluation method. They illustrate this performance through a higher Pearson correlation with the FinSim score of word pairs, which is indexed by finance experts. Taken as a whole, the superiority of deep

learning approaches in investor sentiment classification indicates that these approaches should be adopted as a more powerful decision support system by investors and researchers in finance. Although these word embeddings show outstanding performance in capturing semantic and syntactic similarities of the finance context, other resources are available to extract highly reliable word embeddings that can well represent finance context similarities. It is worth mentioning that there is a trade-off between computational efficiency and accuracy, such that accurate models require higher computation effort. However, the development of high-speed GPUs has significant reduced the time required to build, train, and test DNNs over a reasonable amount of data.

There are a number of methodological limitations that propose new lines for further investigation. First, we trained and validated the classification models over an imbalanced dataset setting a variable class-dependent cost of misclassification for the observations. However, we recommend to test other mechanisms for dealing with the skewness of training dataset, as the threshold-moving approach. Second, we tested n-grams, emojis and emoticons, and negation in developing a valid baseline of traditional investor sentiment classification. The rule-based feature engineering mechanisms will help to capture domain-specific properties of texts from StockTwits to some degree and lead to the development of a robust baseline for DNNs in future studies. Third, the DNNs with more complex topologies might be considered to see if they can improve the classification performance, since they will definitely lead to a higher computation cost. Moreover, combining CNN with an RNN is another option to boost the classification accuracy that enables the model to capture both local features and long-term dependencies of complicated texts from StockTwits. Forth, the dataset for training domain-specific word embeddings is limited to unlabeled messages posted on StockTwits with a small corpus size. Finally, we have created a query dataset of similar finance words including the limited number of highly frequent words in StockTwits. In order to have a robust intrinsic evaluation method, we recommend extending the query dataset and including less-frequently occurring word pairs.

6. References

Al Nasseri, A., Tucker, A., and de Cesare, S. (2014). *Big Data Analysis of StockTwits to Predict Sentiments in the Stock Market*, pages 13–24. Springer International Publishing.

Antweiler, W. and Frank, M. Z. (2004). Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Baker, M. and Wurgler, J. (2000). The equity share in new issues and aggregate stock returns. *The Journal of Finance*, 55(5):2219–2257.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Black, F. (1986). Noise. *The Journal of Finance*, 41(3):528–543.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Boughorbel, S., Jarray, F., and El-Anbari, M. (2017). Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6):e0177678.

Brace, I. (2008). *Questionnaire design: How to plan, structure and write survey material for effective market research*. Kogan Page Publishers, London, UK, 2 edition.

Chan, S. W. and Chong, M. W. (2017). Sentiment analysis in financial texts. *Decision Support Systems*, 94:53 – 64.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Cochrane, J. H. (2000). New facts in finance. *Economic Perspectives. Federal Reserve Bank of Chicago*, 23(3):36–58.

Deng, S., Sinha, A. P., and Zhao, H. (2017). Adapting sentiment lexicons to domain-specific social media texts. *Decision Support Systems*, 94:65 – 76.

Dougal, C., Engelberg, J., GarcÃŋa, D., and Parsons, C. A. (2012). Journalists and the stock market. *The Review of Financial Studies*, 25(3):639–679.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.

García, D. (2013). Sentiment during recessions. *The Journal of Finance*, 68(3):1267–1300.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, Santiago, Chile. IEEE Computer Society.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665, Baltimore, MD, USA. Association for Computational Linguistics.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of ACM*, 60(6):84–90.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Li, J., Chen, X., Hovy, E., and Jurafsky, D. (2015). Visualizing and understanding neural models in NLP. *arXiv preprint arXiv:1506.01066*.

Li, Q. and Shah, S. (2017). Learning stock market sentiment lexicon and sentiment-oriented word vector from stocktwits. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 301–310, Vancouver, BC, Canada. Association for Computational Linguistics.

Li, T., van Dalen, J., and van Rees, P. J. (2018). More than just noise? Examining the information content of stock microblogs on financial markets. *Journal of Information Technology*, 33(1):50–69.

Loughran, T. and McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1):35–65.

Luong, T., Kayser, M., and Manning, C. D. (2015). Deep neural language models for machine translation. In *Proceedings of the 19th Conference on Computational Natural Language Learning*, pages 305–309, Beijing, China. Association for Computational Linguistics.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representation Workshop Track*, pages 1–12, Scottsdale, AZ, USA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, NV, USA. Curran Associates Inc.

Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22 – 31.

Nofsinger, J. R. (2005). Social mood and financial economics. *Journal of Behavioral Finance*, 6(3):144–160.

Novak, P. K., Smailović, J., Sluban, B., and Mozetič, I. (2015). Sentiment of emojis. *PloS ONE*, 10(12):1–22.

Oh, C. and Sheng, O. (2011). Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement. In *Proceediings of Thirty Second International Conference on Information Systems*, pages 1–19, Shanghai, China. Association for Information Systems.

Oliveira, N., Cortez, P., and Areal, N. (2016). Stock market sentiment lexicon acquisition using microblogging data and statistical measures. *Decision Support Systems*, 85:62–73.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods*

in Natural Language Processing, pages 79–86, Philadelphia, PA, USA. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., and Mozetič, I. (2015). The effects of twitter sentiment on stock price returns. *PLoS ONE*, 10(9):1–21.

Reimers, N. and Gurevych, I. (2017). Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348. Association for Computational Linguistics.

Renault, T. (2017). Intraday online investor sentiment and return patterns in the u.s. stock market. *Journal of Banking & Finance*, 84:25 – 40.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics.

Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.

See-To, E. W. K. and Yang, Y. (2017). Market sentiment dispersion and its effects on stock return and volatility. *Electronic Markets*, 27(3):283–296.

Shirani-Mehr, H. (2014). Applications of deep learning to sentiment analysis of movie reviews. Technical report, Stanford University.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Singer, E. (2002). The use of incentives to reduce nonresponse in household surveys. In Robert, M. G., Don, A. D., John, L. E., and Little, R. J. A., editors, *Survey Nonresponse*, pages 163–177. John Wiley & Sons, Ltd., New York, NY, USA.

Sun, F., Belatreche, A., Coleman, S., McGinnity, T. M., and Li, Y. (2014). Pre-processing online financial text for sentiment classification: A natural language processing approach. In *2014 IEEE Conference on Computational Intelligence for Financial Engineering Economics*, pages 122–129, London, United Kingdom. IEEE Computer Society.

Tetlock, P. C., Saar-Tsechansky, M., and Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467.

Vidakovic, B. (2013). *Engineering Biostatistics: An Introduction using MATLAB and WinBUGS*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd.

Wang, T., Wang, G., Wang, B., Sambasivan, D., Zhang, Z., Li, X., Zheng, H., and Zhao, B. Y. (2017). Value and misinformation in collaborative investing platforms. *ACM Transaction on the Web*, 11(2):8:1–8:32.

Wang, X., Jiang, W., and Luo, Z. (2016). Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016): Technical Papers*, pages 2428–2437, Osaka, Japan. The COLING 2016 Organizing Committee.

Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

Yin, W., Kann, K., Yu, M., and Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*.

Zhang, Y. and Wallace, B. (2017). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, volume 1, pages 253–263, Taipei, Tiwan. Asian Federation of Natural Language Processing.

Table 1: Statistics of collected messages from StockTwits.com.

	Volume (Proportion)
Positive	9,161,337 (14.39%)
Negative	2,173,180 (3.41%)
Unlabeled	52,313,016 (82.20%)
Total	**63,647,533 (100.0%)**

Table 2: Examples of word pairs with FinSim index and cosine similarities taken from word embeddings.

Word Pairs	FinSim Index	GloVeST	GloVe	Word2VecST	Word2Vec
Bearish & Negative	0.8	0.6202	0.4193	0.4000	0.4869
Bullish & Positive	0.8	0.6527	0.4551	0.3676	0.5112
Bought & Long	0.7	0.8347	0.5936	0.7090	0.5211
Scalp & Swing	0.75	0.7765	0.0114	0.6385	0.0911
Mutual & Reciprocal	0.2	-0.0855	0.4390	-0.0181	0.5800
Hedge & Mutual	0.75	0.5088	0.3813	0.5276	0.0991

Table 3: Chosen parameter values in feature extraction process.

Feature Extraction	Value[11]
Minimum Document Frequency	5
Maximum Document Frequency	75%
Feature Transformation	Binary

Table 4: Brief info about general word emebeddings and parameter setup to train domain-specific embeddings, GloVeST and Word2VecST.

	GloVe	***Word2Vec***	***GloVeST and Word2VecST***
Unique Tokens	400,000	1,000,000	263,306
Vector Dimension	300	300	300
Number of Tokens	6,000,000,000	100,000,000,000	838,009,514

Table 5: Parameter setup for CNN, GRU, and LSTM.

Common parameters	
Maximum Length	30
Unknown Embedding Vector	$U[-1, 1]$
Kernel and bias Initializer	Normal (He et al., 2015)
Loss Function	Binary Cross-entropy
Batch Size & Epoch	100 & 100

CNN		***LSTM***		***GRU***	
Filter Size	250	Recurrent Layers	2	Recurrent Layers	3
Kernel Size	[4, 3, 4]	Recurrent Unit	125	Recurrent Unit	75
Pooling	Max-pooling	Recurrent Dropout	0.01	Recurrent Dropout	0.02
Dropout	0.49	Dropout	0.42	Dropout	0.27
Activation	$tanh$	Activation	$tanh$	Activation	$tanh$
Optimizer	Adadelta	Optimizer	Adadelta	Optimizer	Adadelta

Figure 1: Downscaled Convolutional Neural Network (CNN) for investor sentiment classification.

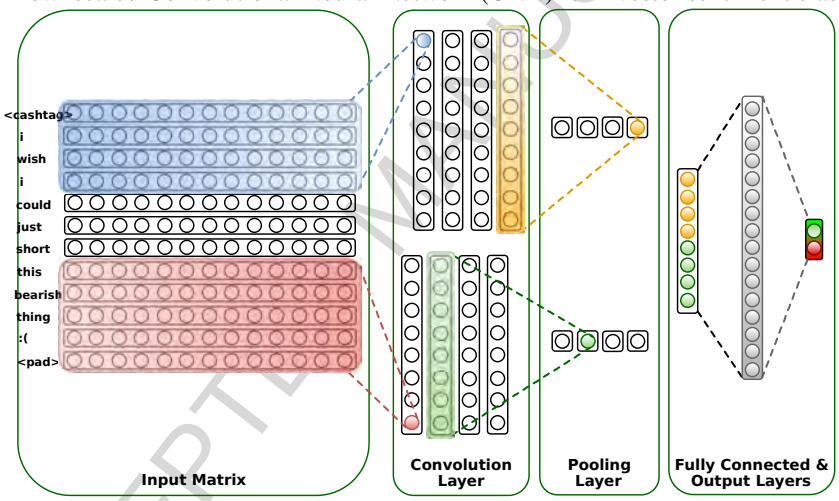
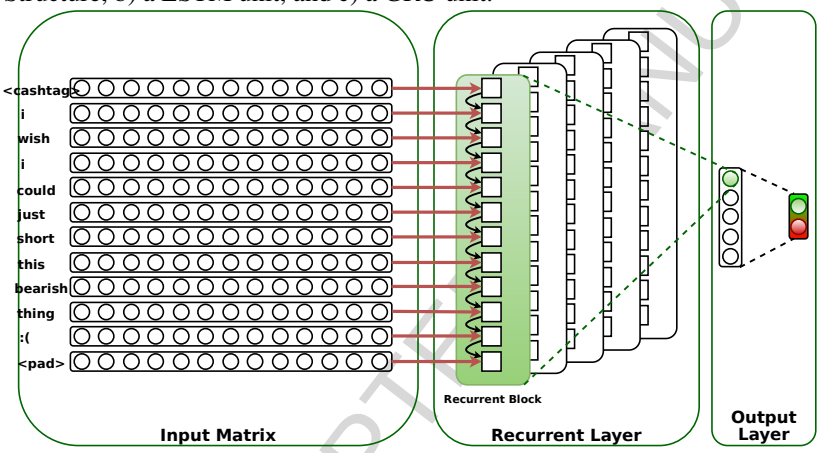
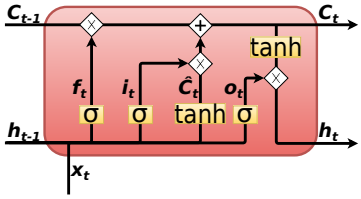


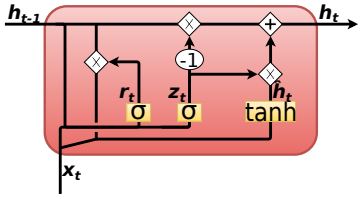
Figure 2: Downscaled Recurrent Neural Networks implemented on investor sentiment classification. a) RNN Structure, b) a LSTM unit, and c) a GRU unit.



(a) The RNN Architecture.



(b) Long Short-Term Memeory.



(c) Gated Recurrent Unit.

Figure 3: Performance of various classifiers on different feature types.

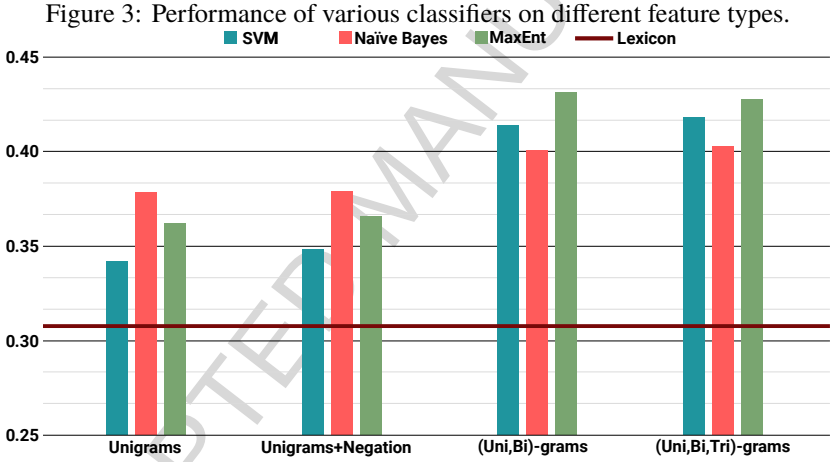
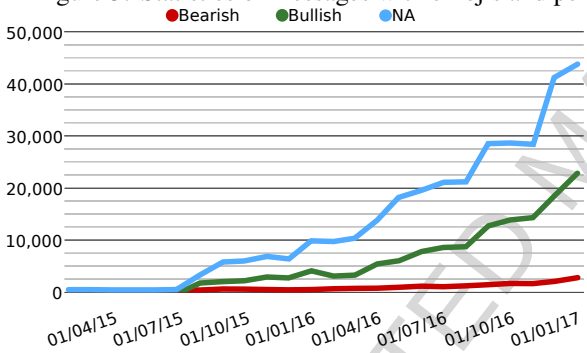


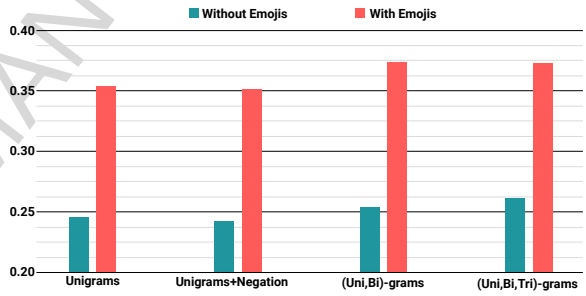
Figure 4: Emoji cloud for popular emojis in a) bullish messages and b) bearish messages.

(a) (b)

Figure 5: Statistics of messages with emojis and performance of MaxEnt with emojis and without emojis.



(a) Frequency of messages with emojis.



(b) Performance of the classifier with and without emojis.

Figure 6: FinSim index versus cosine similarity of pairs.

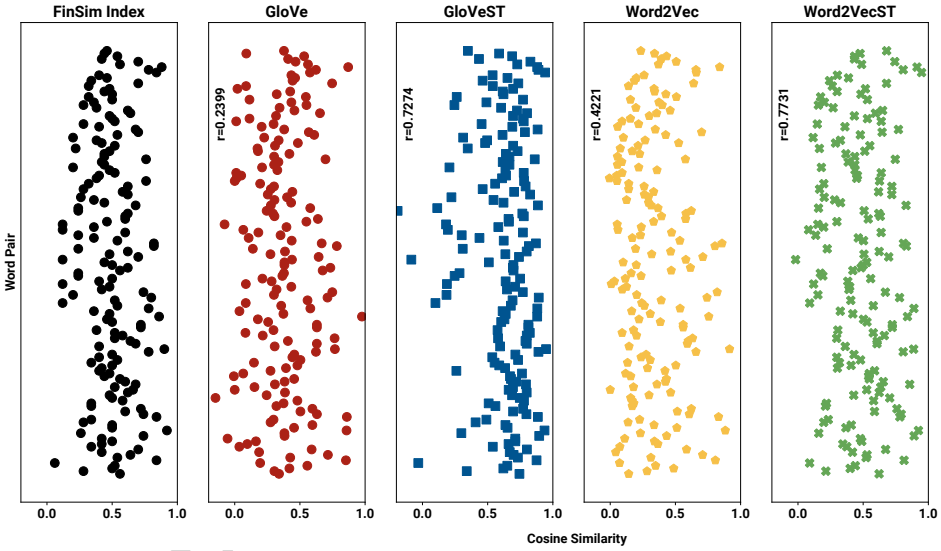
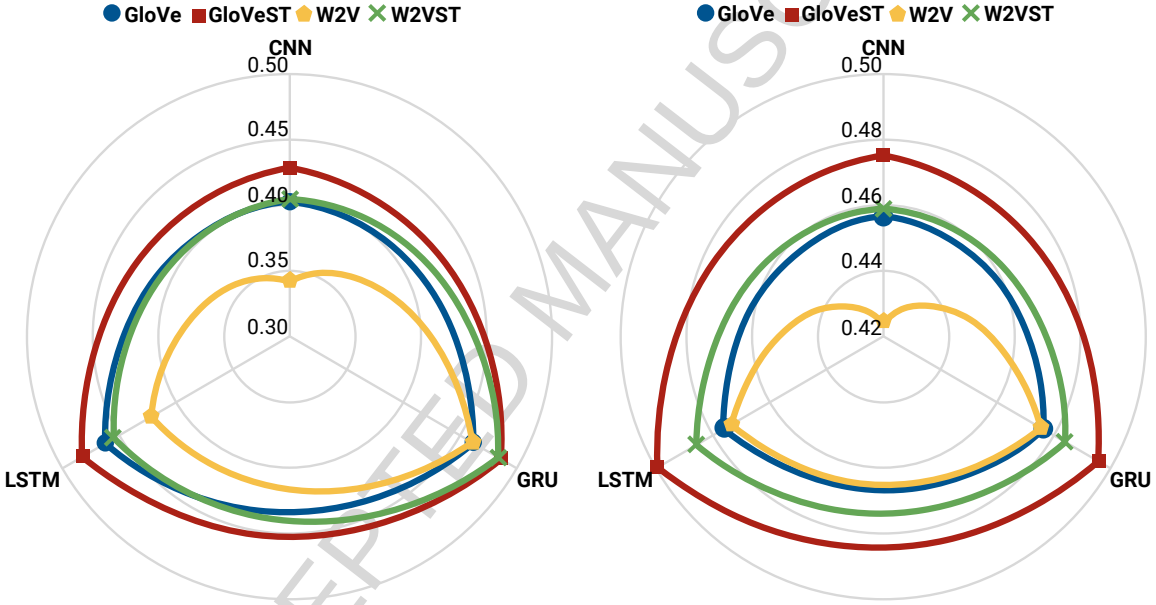


Figure 7: Performance of Deep Learning Algorithms with Static and Non-static Word Embeddings.



(a) Static

(b) Non-static

Figure 8: Location[12]of sentiment related words before (black) and after (positives & negatives).

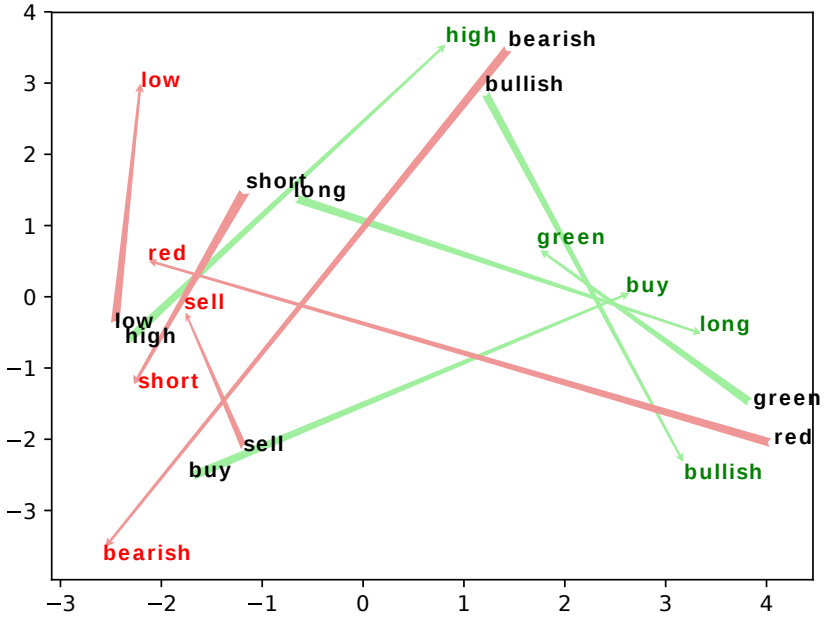
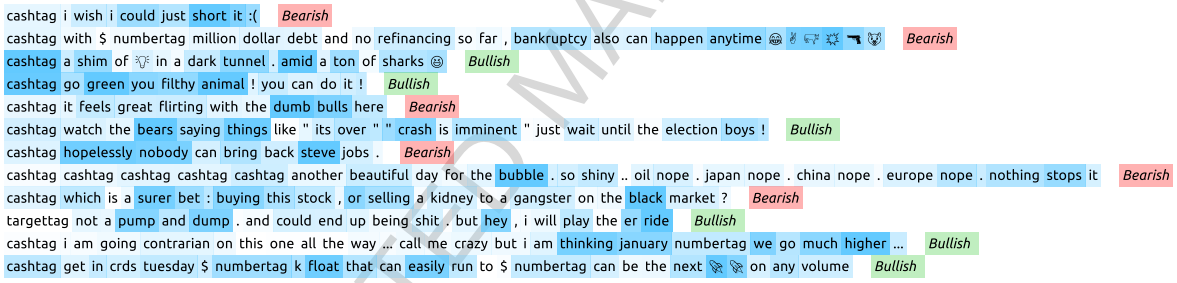


Figure 9: Saliency concentration over every token in the sentences extracted from LSTM with GloVeST (the best combination). Darker shadows show intense saliency concentration.



Mr. Nader Mahmoudi

Nader is a Ph.D. candidate in Finance at the University of Newcastle since 2016. His research focuses on data-driven approaches and business applications with expertise in statistical learning, text mining, and classification. In his Ph.D. dissertation, Nader aims to explore the challenges in measurement of investor sentiment from social networks and propose effective and innovative solutions for it. He has received his M.Sc. in Industrial Engineering from Özyeğin Uinversity, Istanbul, Turkey. During this period, Nader joined a nation-wide research project granted by Scientific & Technological Research Council of Turkey (TÜBİTAK). He has presented his research outcomes in the different journals and conferences.

Dr. Paul Docherty

Paul is an academic in finance who joined Monash in July 2017 after ten years at the University of Newcastle. His main research interest is in empirical asset pricing, where he has published 25 peer-reviewed articles in journals including the Journal of Financial and Quantitative Analysis, the International Review of Financial Analysis and Accounting and Finance. He has supervised eight PhDs and seven Honours students to completion. Paul's teaching tends to align with this research interest. He has won several awards for teaching, including the NUPSA award for 'Teacher of the Year' and a Vice-Chancellor's citation for teaching excellence at the University of Newcastle.

Professor Pablo Moscato

Australian Research Council Future Fellow Prof. Pablo Moscato is the founding co-director of the Priority Research Centre for Bioinformatics, Biomarker Discovery and Information-based Medicine

(2006-Present). His main research interest is in Data Science and Artificial Intelligence, where he has authored and co-authored more than 200 academic publications including books, book chapters, refereed journal articles, and conference papers. His collaborations span all of The University of Newcastle's faculties. His most recent activities, involving partnership with the Faculty of Business and Law, have included the analysis of online consumer behaviours and online customer brand engagement. He has coordinated and lectured more than 10 courses aligned with his research interests since 1986.

Highlights

- Emojis significantly improve investor sentiment classification accuracy.
- Deep neural networks (DDNs) outperform traditional classification algorithms.
- New method developed to assess word embeddings in a domain-specific way.
- Domain-specific word embeddings better capture investor sentiment.

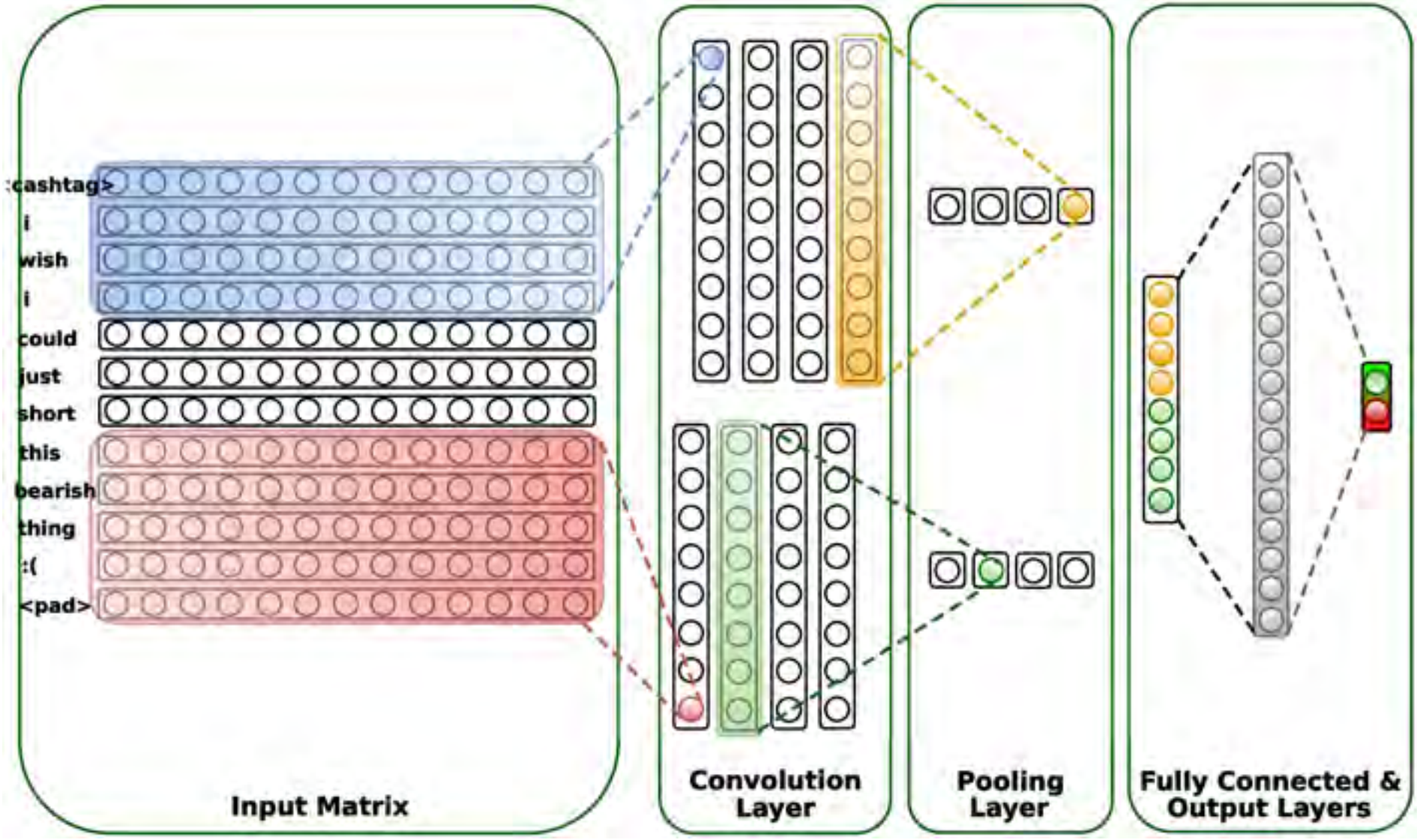

:cashtag>
i
wish
i
could
just
short
this
bearish
thing
:(
<pad>
Input Matrix
Convolution Layer
Pooling Layer
Fully Connected & Output Layers

Figure 1

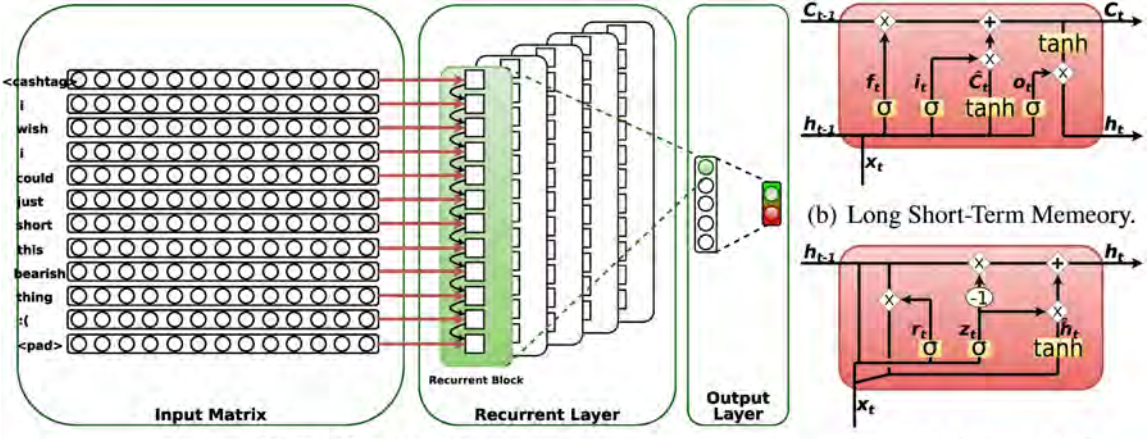

<cashtag>
i
wish
i
could
just
short
this
bearish
thing
:(
<pad>
Input Matrix
Recurrent Block
Recurrent Layer
Output Layer
(a) The RNN Architecture.
(b) Long Short-Term Memeory.
(c) Gated Recurrent Unit.

Figure 2

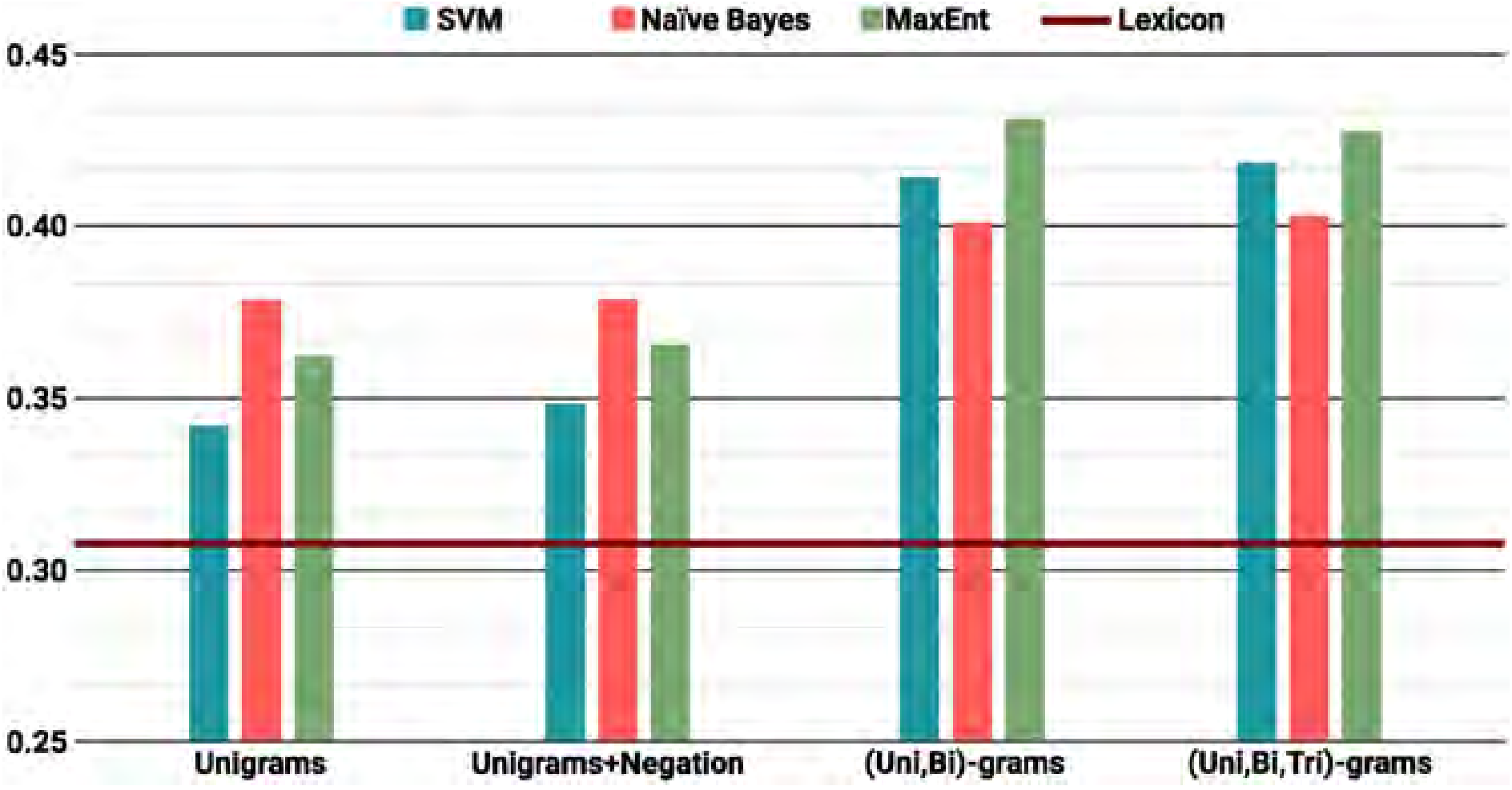
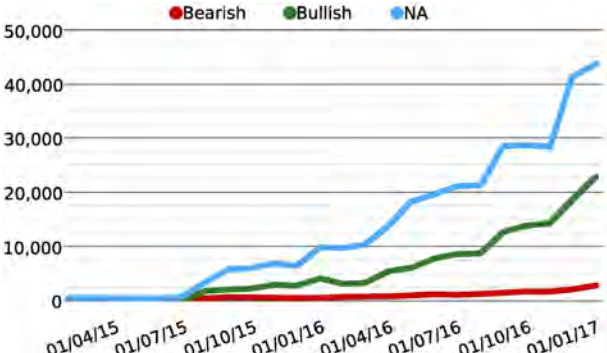
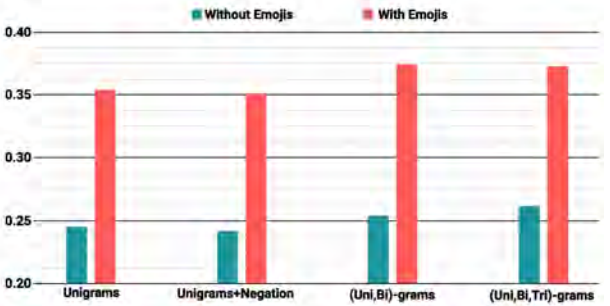

SVM
Naïve Bayes
MaxEnt
Lexicon
0.45
0.40
0.35
0.30
0.25
Unigrams
Unigrams+Negation
(Uni,Bi)-grams
(Uni,Bi,Tri)-grams

Figure 3

Figure 4



(a) Frequency of messages with emojis.



(b) Performance of the classifier with and without emojis.

Figure 5

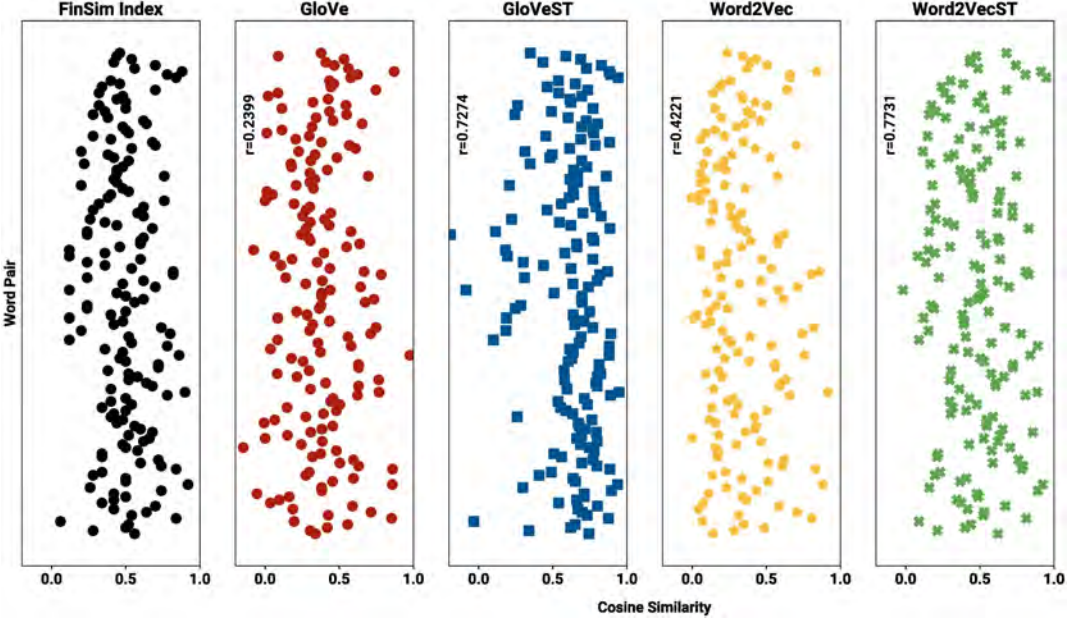
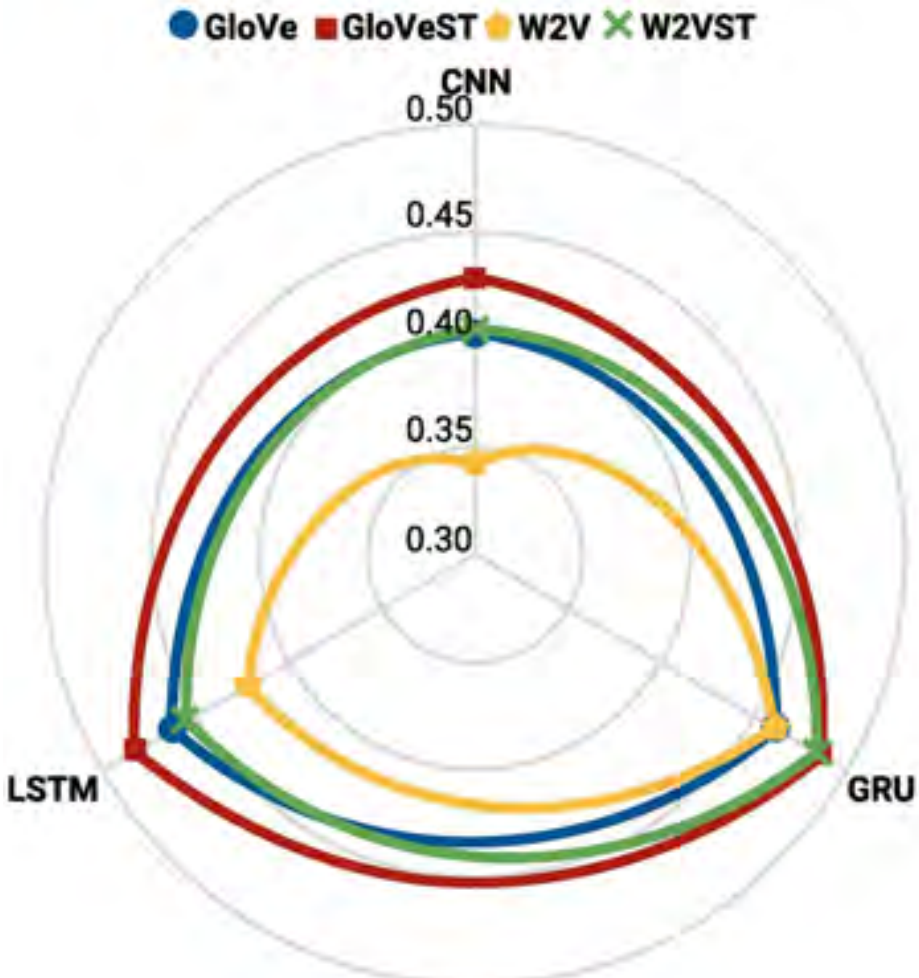
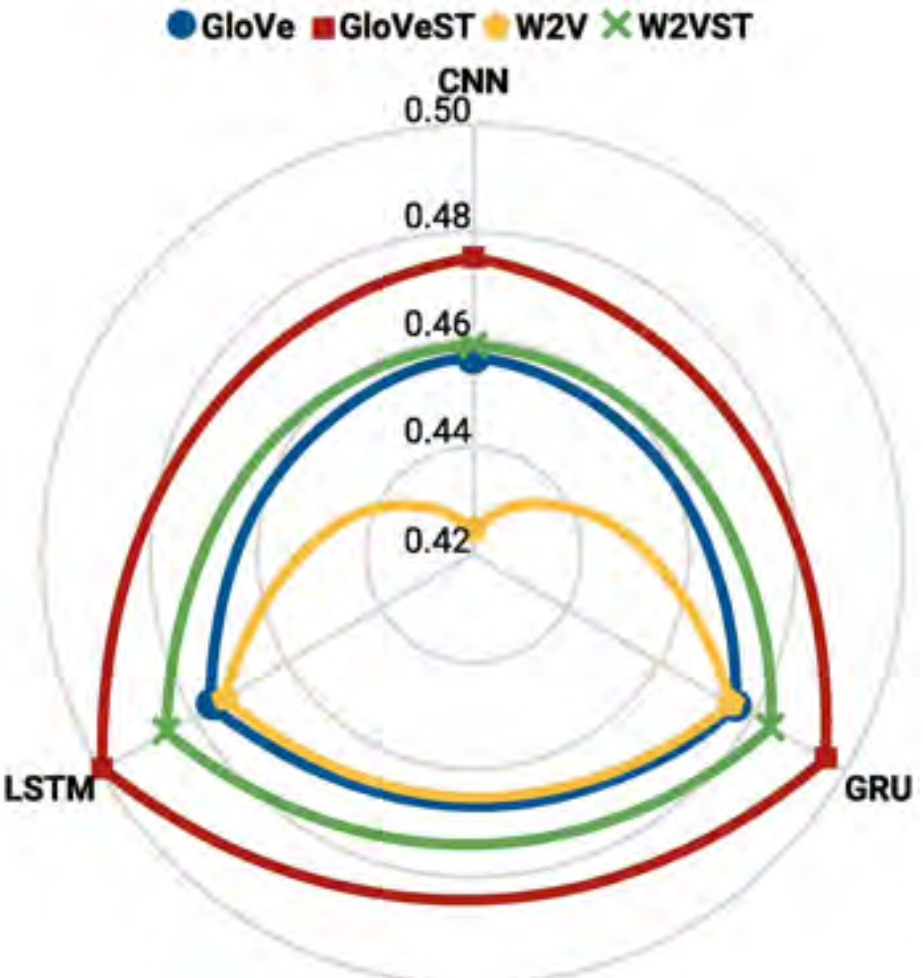

FinSim Index
GloVe
GloVeST
Word2Vec
Word2VecST
r=0.2399
r=0.7274
r=0.4221
r=0.7731
Word Pair
0.0
0.5
1.0
Cosine Similarity

Figure 6



(a) Static



(b) Non-static

Figure 7

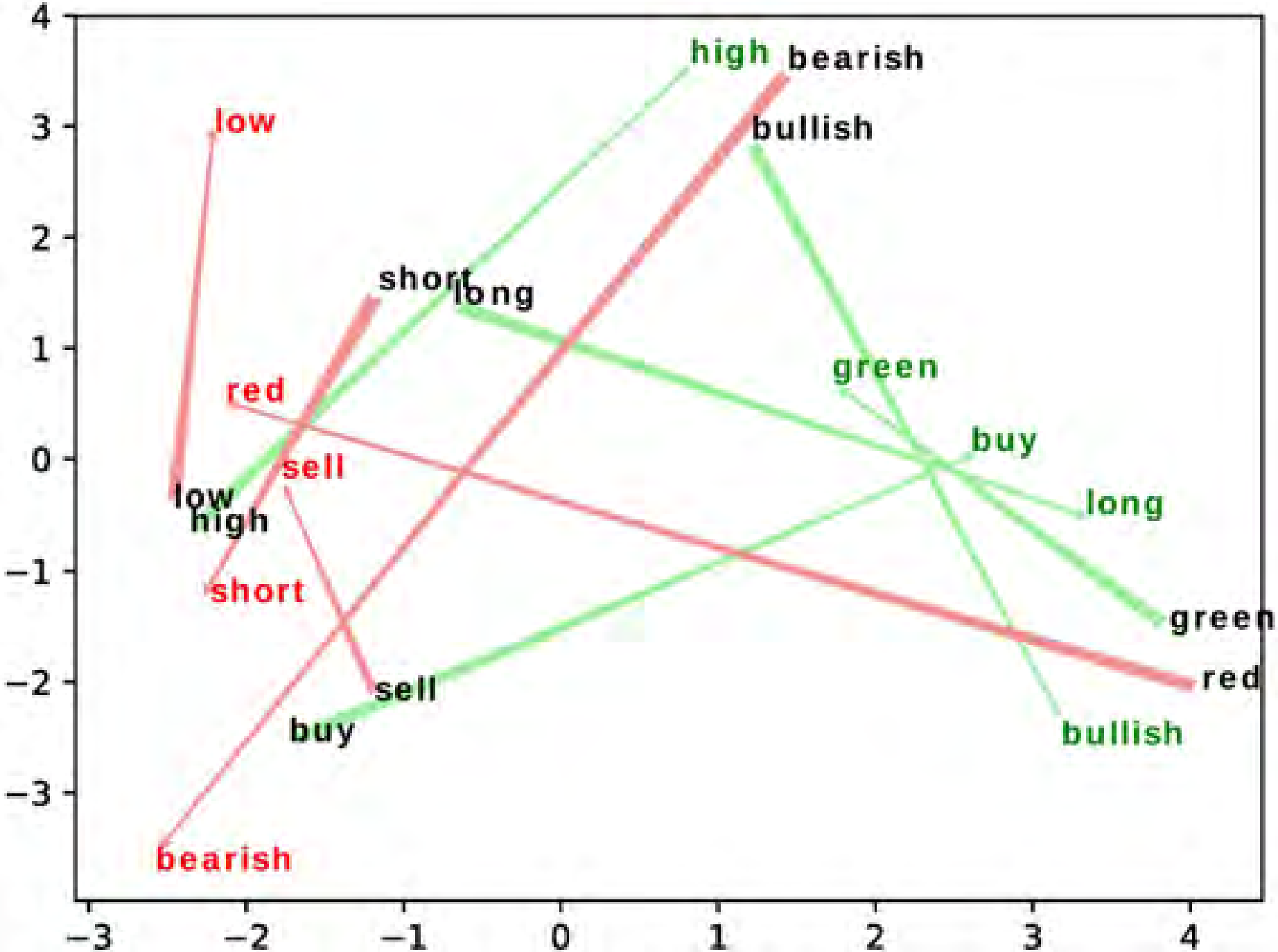

high
bearish
low
bullish
short
long
green
red
buy
sell
low
high
long
short
green
red
sell
buy
bullish
bearish
4
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
4

Figure 8

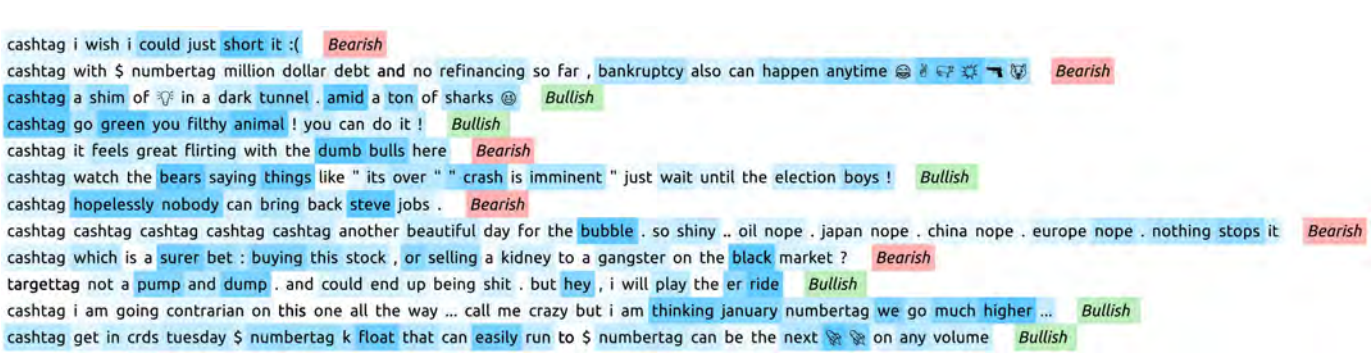

cashtag i wish i could just short it :(Bearish
cashtag with $ numbertag million dollar debt and no refinancing so far , bankruptcy also can happen anytime Bearish
cashtag a shim of in a dark tunnel . amid a ton of sharks Bullish
cashtag go green you filthy animal ! you can do it ! Bullish
cashtag it feels great flirting with the dumb bulls here Bearish
cashtag watch the bears saying things like " its over " " crash is imminent " just wait until the election boys ! Bullish
cashtag hopelessly nobody can bring back steve jobs . Bearish
cashtag cashtag cashtag cashtag cashtag another beautiful day for the bubble . so shiny .. oil nope . japan nope . china nope . europe nope . nothing stops it Bearish
cashtag which is a surer bet : buying this stock , or selling a kidney to a gangster on the black market ? Bearish
targettag not a pump and dump . and could end up being shit . but hey , i will play the er ride Bullish
cashtag i am going contrarian on this one all the way ... call me crazy but i am thinking january numbertag we go much higher ... Bullish
cashtag get in crds tuesday $ numbertag k float that can easily run to $ numbertag can be the next on any volume Bullish

Figure 9