World Wide Web (2022) 25:513–531
https://doi.org/10.1007/s11280-021-00993-1

Leveraging statistical information in fine-grained financial sentiment analysis

Han Zhang[1] · Zongxi Li[2] · Haoran Xie[3] · Raymond Y. K. Lau[4] · Gary Cheng[5] · Qing Li[6] · Dian Zhang[7]

Received: 21 July 2021 / Revised: 29 September 2021 / Accepted: 21 December 2021 /
Published online: 5 February 2022

Abstract

The recent development of deep learning-based natural language processing (NLP) methods has fostered many downstream applications in various fields. As one of the applications in the financial industry, fine-grained financial sentiment analysis (FSA) aims to understand the sentimental orientation, i.e., bullish or bearish, of financial texts by predicting the polarity score and has been widely applied in the financial industry stock-related opinion mining. Because of the lack of a large-scale labeled dataset and the domain-dependent nature, FSA is challenging. Previous works mainly focus on constructing and exploiting handcrafted lexicons that encode expert knowledge to enhance the semantic features in decision making, which yields improvements but are expensive to acquire. This paper proposes a lightweight regression model incorporating the statistical distribution of a term over the polarity range, say between −1 and 1, to address the fine-grained FSA task. More concretely, we first count each word's appearance at different polarity intervals and produce a statistic-based representation for each text, which will be encoded as a corpus-level statistical feature vector by an autoencoder. Subsequently, the obtained feature vector will be integrated with the semantic feature vector in the regression model. Our experiments show such a model can produce significant improvements compared with the baseline models on two FSA subsets, i.e., news headlines and microblogs, without a computational overhead. Furthermore, we notice the signs that lexicon-based approaches have neglected can play an important role in FSA.

Keywords Financial sentiment analysis · Sentiment analysis · Natural language processing · Information retrieval

This article belongs to the Topical Collection: *Special Issue on Web Intelligence = Artificial Intelligence in the Connected World*
Guest Editors: Yuefeng Li, Amit Sheth, Athena Vakali, and Xiaohui Tao

✉ Zongxi Li
zoli@hkmu.edu.hk

Extended author information available on the last page of the article.

1 Introduction

With the advance of the NLP technique, public opinion mining over online user-generated textual content, such as news and blogs, has gained great attention from the financial industry. Understanding sentimental orientation from online texts helps to investigate the investors' opinion towards the overall stock market or certain stock and facilitates the modeling of the financial market dynamics and stock forecasting [27, 28]. Therefore, FSA is an important research topic of financial technology (FinTech) and has long been the tradition of trading practice [47]. The objective of FSA is to classify a piece of financial text as expressing bullish or bearish opinions toward certain arguments [46]. Fine-grained FSA aims at predicting the exact sentiment score of a given financial text. The sentiment scores are floating-point values in the range of -1 (very negative/bearish) to 1 (very positive/bullish), with 0 designating neutral sentiment [8].

Although extensively discussed in past research, sentiment analysis-relevant topics in a specific domain [7], such as FSA, are still challenging tasks. Because annotating requires expert knowledge, acquiring large-scale datasets for model training is expensive. As a consequence, the models achieving good performance in the general domain suffer data sparsity issue in FSA, which is referred to as a problem of domain adaptation [46]. To address such an issue, researchers have incorporated various hand-engineered features to boost the model performance, such as sentiment lexicons (e.g., SenticNet [6], Vader [17], and NRC [37]), opinion lexicon [16], hashtag lexicon [36], etc. A summary of lexicon applications at SemEval 2017 Task 5 (Fine-grained Financial Sentiment Analysis Challenge) can be found at [8]. Since these lexicons present different aspects at different scales, ensemble learning methods are widely adopted to integrate features from multiple lexicon resources [1, 18, 20]. Although lexicon-based methods show improvements on sentiment analysis tasks, they still suffer from the following limitations: (1) existing lexicon resources are mainly from the general domain, which may not be compatible with FSA; (2) non-text patterns that are prominent in FSA are neglected by the lexicon-based methods. We notice that the semantic patterns of provided spans in FSA, which are the list of strings from the message expressing sentiment, are different from the general domain sentiment analysis. Particularly, in microblogs, a significant proportion of spans is of the similar patterns with the examples in Table 1, where no explicit sentiment is expressed by any terms in the span.

Recent works have discovered that some primitive corpus-level features, such as word frequency and distribution over labels, are overlooked by the current deep learning paradigm, and such features could be as informative as lexicons [25, 26, 51]. Different from the handcrafted lexicons, statistical information is an intrinsic attribute of a corpus and is easy to retrieve. Using the statistical feature in information retrieval is not a new thing, and the most representative method is the *term frequency-inverse document frequency* (TFiDF), which is a straightforward approach for document modeling. The motivation behind using

Table 1 Examples where lexicon-based method can hardly extract effective sentiment patterns

Example 1
\$ CAT +5.10%, \$ RIO +4.54%, \$ FCX +3.53%, \$ FXI +2.93%,
\$ BHP +3.04%, \$ YHOO +2.61%, \$ X +2.27% ...
Example 2
2013 LONGS(12/31/2012 close), \$ CLWR > 2.89, \$ SIRI > 2.89,
\$ SWHC > 8.44, \$ GE > 20.99, I'm UP(+8.332%) YOU?

statistical features in supervised learning is elegant: we hope to get a task-related representation that highlights more discriminative words or terms in the underlying task. Intuitively, in sentiment analysis, the distribution of a word appearing in both positive and negative instances is a natural indicator of its sentiment orientation, an ideal alternative of lexicons. Moreover, the statistical pattern of signs' usage, such as $+$, $-$, $>$, and $<$, can also be leveraged. The current strategies to model the term's frequency are similar: Li et al. [25] define the *term-count-of-labels* (TCoL) and utilize the term's occurrence explicitly; Zubiaga [51] exploits the *term frequency-category ratio* with a designated weighting scheme. Though, these works address classification tasks, while in a fine-grained FSA, we have continuous sentiment scores instead of discrete categories. In this paper, we propose an efficient method to incorporate statistical features in a regression task. To model the term's distribution over different polarity intensities, we partition the polarity range into five polarity intervals, i.e., very negative (-1.0 to -0.6), negative (-0.6 to -0.2), neutral (-0.2 to 0.2), positive (0.2 to 0.6), and very positive (0.6 to 1.0), and count the occurrences of each term in each interval. A statistical representation of each text instance is constructed by concatenating each term's statistical distribution vector. The discrete representation will be encoded to a continuous space by an autoencoder, as Li et al. [25] have proven that a variational encoding component can enhance the robustness of using statistical information. The encoded representation will be incorporated in the regression model by an explicit concatenation operation with extracted semantic representation. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to leverage corpus-level statistics explicitly in a regression model and prove it as an effective approach.
- In this paper, we examine the statistics of signs in microblogs and their role in enhancing regression performance.
- We conduct extensive experiments on news headline datasets and microblog datasets in the financial field. The results show that our proposed method produces significant improvements on baseline models without a computational overhead.

2 Related work

Although investors in the market are assumed to be rational according to the Efficient Market Hypothesis (EMH) which states that securities prices in the efficient market fully reflect all publicly available information [11], market sentiment in the practice impacts stock prices as well [34]. Researchers have found that the market sentiment is positively associated with contemporaneous and future stock returns [40], and helps predict market volatility and trading volume [2]. Financial texts are playing an increasingly important role in measuring market sentiment due to the proliferation of social media platforms. The sentiment information derived from social media such as Twitter and StockTwits is significantly correlated with stock risk in a short term [49]. Therefore, some trading strategies are designed on the basis of textual financial news and have achieved significant returns [12]. From the perspective of corporate governance, the emotions in the financial text effect the firm's decision-making. For example, investors' surprise emotion has a significantly negative effect on firm's post-M&A stock returns, so FSA can help investors to make better investment decision [45].

Among NLP areas, sentiment analysis is one of the most important tasks [29, 30]. Do et al. [10] classified the study of sentiment analysis into three levels: document, sentence, and aspect. Mowlaei et al. [38] proposed statistical methods and a genetic algorithm to improve the lexicon generation methods for aspect-based problems. Mai and Le [33] found that the model with two levels combined can perform better than that using the single level. Xu et al. [48] incorporate context-relevant concepts into convolutional neural networks (CNNs) for short text classification on sentiment analysis datasets. Cai et al. [5] propose an attention-based multi-task learning framework based on recurrent neural networks (RNNs) for sentiment analysis. However, compared to traditional sentiment analysis, financial sentiment analysis is more challenging because of the features of the financial text. Firstly, the sentiment in the financial text reflects the market participants' expectations on the near-term market [4]. While traditional sentiment analysis, which emphasizes the consumers' current feelings, is often used in the user-product scenarios, FSA works in predicting the stock prices and monitoring the abnormal returns. For example, the negative sentiment expressed in an investor's message indicates the investor may doubt the market in the near future. Secondly, the financial text is more implicit in the sentiment contained, because the emotion words such as "buy" "sell" and "exciting" in the financial text are not diverse, and the financial text usually contains various technical terms and expert knowledge along with numerous statistics [1]. Thirdly, due to various social media platforms and different forms of the financial text, the sentiment analysis needs to be considered from miscellaneous perspectives such as macroeconomic information, microstructure factors, event-oriented, company-specific [32].

In the previous research, the main methodologies used in FSA include lexicon-based approach, regular machine learning approach and deep learning such as CNN, RNN, LSTM, and attention mechanism. In the early stage of FSA, the General Inquirer (GI) built-in dictionary is the most widely used word list [42]. Loughran and McDonald [31] found that almost three-fourths (73.8%) of the negative word counts in the word list are attributable to words that are typically not negative in a financial text. Hence, the authors developed alternative finance-specific word lists. Despite the good performance of using domain-specific dictionaries such as Loughran-McDonald financial sentiment dictionary (LMFSD), the lexicon-based approach is easy to miss the critical information in the fine-grained financial sentiment analysis. Li [24] applied the Naïve Bayesian machine learning algorithm to examine the forward-looking statements in the corporate filings and demonstrated that the regular machine learning approach could achieve better performance than the lexicon-based approach. Wang et al. [44] classified StockTwits tweets as "bullish" or "bearish" by applying machine learning approaches and found that the SVM model was the most accurate among Naïve Bayes, SVM, and Decision Tree. However, the problem of the regular machine learning approach is that it fails to extract the complex features in a long sentence. After [21] firstly proposed a classic TextCNN model to extract features for sentiment classification, other researchers applied CNN on model document-level [43], character-level [50], and word-level [19]. Because its limitation on handling the long dependency of sequential input financial text, LSTM [15], BiLSTM [14] and BiGRU [3] are employed in FSA. Despite the good effects on sentiment analysis, these deep learning models pay little attention to some discriminative words. We propose to merge additional statistical information to avoid noise to the classifier.

3 Methodology

The overall framework of the proposed approach is depicted in Figure 1. In this section, we introduce the framework in terms of its components.

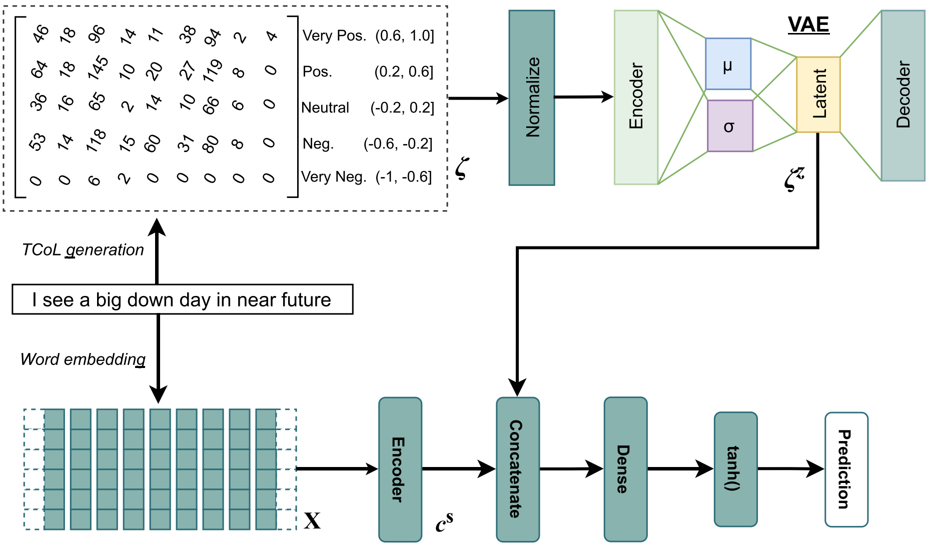

Figure 1 The generic framework of the proposed method. Given a sentence, our model first generates a TCoL representation of the sentence and encodes the statistical feature into latent vector $\boldsymbol{\zeta}^{\mathbf{z}}$ via a variational autoencoder. The generated statistical representation $\boldsymbol{\zeta}^{\mathbf{z}}$ is concatenated with latent semantic feature vector in the regression model

3.1 Statistical information

In this work, we define and leverage the statistical information following Li et al. [25]. Li et al. [25] define the statistics of terms towards labels as the *term-count-of-labels* (we will adopt the same notion in this paper):

Definition 1 Given a word w and a set of labels of c classes, the term-count-of-labels (TCoL) vector of w is

$$\boldsymbol{\zeta}^{w} = [\zeta_1, \ldots, \zeta_c], \tag{1}$$

where ζ_i is the count of word w on label i. Given a sentence $s = \{w_i\}_{i=1}^{m}$, the TCoL matrix of sentence s is

$$\boldsymbol{\zeta}^{s} = \left[\boldsymbol{\zeta}^{w_1}, \ldots, \boldsymbol{\zeta}^{w_m}\right]. \tag{2}$$

In the fine-grained FSA that we address, the goal is to predict the numerical value of the sentiment score. Thus the categorical labels are not available. An intuitive idea is to manually create *visual* sentiment categories by partitioning the polarity range. Counting the occurrences over the *visual* sentiment classes, the statistic profile of a term can reflect its *conventional* usage at different sentiment levels under a specific scenario. According to the numerical distribution of the sentiment scores in both microblog dataset (Figure 2) and news headline dataset (Figure 3), we define five polarity intervals, i.e., very negative (−1.0 to −0.6), negative (−0.6 to −0.2), neutral (−0.2 to 0.2), positive (0.2 to 0.6), and very positive (0.6 to 1.0), and construct the notion of TCoL based on the partitioned intervals.

The TCoL notion reflects a global distribution on different categories as features of a word, which are highly informative regarding the information retrieval by modeling the word relevance [39, 41]. Intuitively, if a word or term w frequently or barely occurs on all

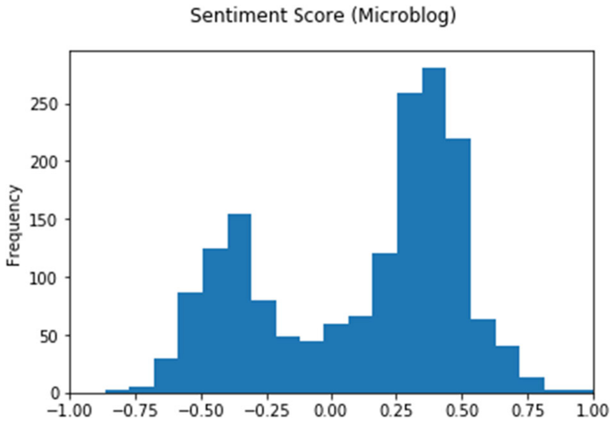

Figure 2 Histogram of sentiment score's distribution in the microblog dataset

categories, we shall assume that w has a limited contribution to the feature selection. In contrast, if a word appears more frequently in the specific sentiment intervals, we can assume this word is discriminative and is used by people to express certain sentiment orientation and intensity. Thus, the TCol representation can be regarded as an appropriate alternative to handcrafted lexicon knowledge. Note that the TCoL dictionary V is obtained from the training set only.

3.2 Variational encoding of TCoL

The TCoL representation of a financial text consists of integer counts of terms. Therefore, the statistical information is not compatible with semantic features in scale and dimension. Furthermore, because of the small-size dataset, the obtained TCoL may deviate from the term's utilization in real life and compromise the performance [25]. To overcome these challenges, we employ an autoencoder to map discrete TCoL vectors into a continuous representation after the normalization process, which is encoded with corpus-level statistics

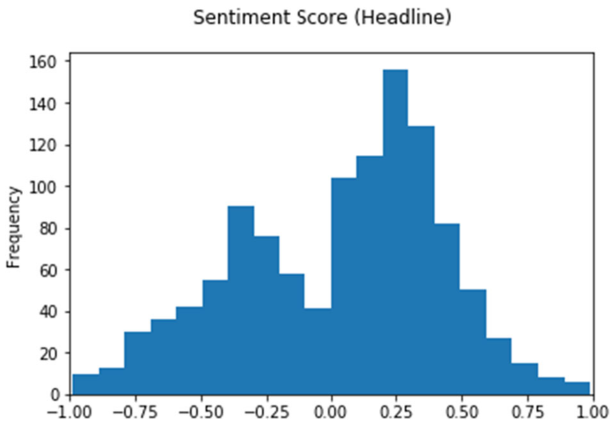

Figure 3 Histogram of sentiment score's distribution in the news headline dataset

and can be regarded as a global representation of each sentence in the dataset. Meanwhile, the encoding and decoding process can help to alleviate errors and noises in the TCoL. Moreover, we find it is beneficial to bound the latent space with a variational layer, so as the encoded representation is still scalable in the afterward modules. Thus, we employ the Variational Autoencoder (VAE) [22] in this paper.

We generate TCoL for all sentences in a dataset at the preprocessing stage. A dictionary is constructed to update the TCoL vector of each term as we iterate through the corpus. Then we concatenate the terms' TCoL vectors in a sentence and obtain the sentence-level TCoL matrix for each textual instance. Finally, we have $\mathbf{Z} = \{\boldsymbol{\zeta}^{s}_{(i)}\}_{i=1}^{N}$ for a given corpus, which consists of N independent and identically distributed (i.i.d.) discrete TCoL variable $\boldsymbol{\zeta}$. To construct a variational encoding model, we consider a generative process that generates all TCoL vectors by a random process from an unobserved continuous hidden variable $\mathbf{z}$. The generative process is of two stages: (1) the hidden variable $\mathbf{z}$ is sampled from a prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$; (2) a TCoL variable $\boldsymbol{\zeta}_{(i)}$ is generated from some conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}_{(i)}|\mathbf{z})$.

The generative process is not visible to us, and the details of parameters and hidden variables are unknown to us. Moreover, because the integral of the marginal likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}) = \int p_{\boldsymbol{\theta}}(\mathbf{z}) p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}|\mathbf{z}) d\mathbf{z}$ is intractable and the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{\zeta})$ is also intractable, we cannot explicitly estimate the generative model parameters $\boldsymbol{\theta}$ and the hidden variables $\boldsymbol{\zeta}$. To solve the above problems, we adopt another recognition model $q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta})$ to approximate the intractable true posterior by inferencing the variational parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ jointly.

Substituting the real posterior with the approximation distribution, we can have the marginal likelihood composed of a sum over the marginal likelihoods of individual $\boldsymbol{\zeta}$:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{\zeta})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\zeta}). \tag{3}$$

We can optimize the inference model by maximizing (3). Since the Kullback–Leibler (KL) divergence term in (3) is non-negative, the likelihood term $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\zeta})$ is the variational lower bound on the marginal likelihood, i.e.,:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\zeta}) = \mathbb{E}_{q_{\boldsymbol{\phi}(\mathbf{z}|\zeta)}} \left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}, \mathbf{z}) \right], \tag{4}$$

which can be rewritten as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\zeta}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta}) \| p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\zeta)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}|\mathbf{z}) \right], \tag{5}$$

where the left-hand side KL term in (5) has a closed-form solution, and the right-hand side expectation term can be considered as the reconstruction error between the original input and the generative output. The reparameterization trick is employed to fit the variational framework into an end-to-end deep learning model: we refer the approximation model $q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta})$ as a probabilistic encoder and the generative process $p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}|\mathbf{z})$ as the probabilistic decoder. As a common approach, we use a multivariate Gaussian as the approximate prior. Therefore, we employ two encoders networks to produce two sets of μ and σ as a prior distribution's mean and standard deviation, respectively, to sample the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta})$ with a diagonal covariance structure:

$$\log q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta}) = \log \mathcal{N}\left(\mathbf{z}; \mu, \sigma^2 \mathbf{I}\right). \tag{6}$$

By optimizing the VAE model, we can encode discrete TCoL input to the latent variables $\boldsymbol{\zeta}^{\mathbf{z}}$ via the probabilistic encoder. The $\boldsymbol{\zeta}^{\mathbf{z}} \in \mathbb{R}^K$ vector will be the global representation of TCoL, where K is the dimension of latent TCoL.

As a side note, the training of the VAE model is an offline process and is independent of the main regression model. The representation $\boldsymbol{\zeta}^{\mathbf{z}}$ is generated during the preprocessing stage.

3.3 Semantic feature extractor

We extract semantic features from textual input and project semantic features into a shared space with statistical representation. The textual input is a financial news headline or a financial microblog s with a fixed length m. The embedding layer first map each word or term in one instance into a k-dimensional continuous space $\mathbf{x}_i$ and form a $k \times m$ matrix $\mathbf{x} = [\mathbf{x}, \cdots, \mathbf{x}_m]$, which will be the input of the semantic feature extraction module. Multiple popular extractors, i.e., TextCNN, BiLSTM, and BiGRU, can be employed to produce latent semantic feature map.

More concretely, for a TextCNN [21] layer, we apply filters $\mathbf{W}^f \in \mathbb{R}^{h \times k}$ with window size h on the embedding matrix $\mathbf{x}$. The extracted feature c_i is generated from a window of embedding vectors $\mathbf{x}_{i:i-h+1}$:

$$c_i = \boldsymbol{f}\left(\mathbf{W}_f \circledast \mathbf{x}_{i:i-h+1} + b\right), \tag{7}$$

where, $b \in \mathbb{R}$ is the bias term, and $\boldsymbol{f}(\cdot)$ is a non-linear function. We apply d filters in total to produce a latent feature map $\mathbf{c} \in \mathbb{R}^{d \times m}$ with padding in the semantic space. A max-over-time pooling operation extracts the most prominent value $\hat{c} = \max(\mathbf{c}_i)$. By doing this, we obtain a latent semantic vector $\mathbf{c}^s = [\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_d]$.

The recurrent models, i.e., BiLSTM [14] and BiGRU [3], summarize the contextual information from both directions of a sequential input. In each LSTM and GRU cell, a gate mechanism continuously updates the hidden state vector $\mathbf{h}_i$ by deciding what information to take in and what information to forget. Under the bidirectional setting, the forward function $\overrightarrow{f}$ reads the sequence from the start to the end and outputs the forward hidden state $\overrightarrow{\mathbf{h}}$, and the backward function $\overleftarrow{f}$ reads the same sequence reversely and outputs the backward hidden state $\overleftarrow{\mathbf{h}}$. We concatenate the forward vector and the backward vector as the latent semantic feature vector $\mathbf{c}^s = [\overrightarrow{\mathbf{h}}; \overleftarrow{\mathbf{h}}]$.

3.4 Regression model

We concatenate the statistical representation $\boldsymbol{\zeta}^{\mathbf{z}}$ obtained in Section 3.2 and the latent semantic feature vector $\mathbf{c}^s$ obtained in Section 3.3 as the input of regression model. The concatenation operation is exploited to combine the encoded statistical knowledge with semantic feature in the lightweight regression process in a straightforward manner. After passing through fully-connected layers, the combined feature vector is mapped into the 1-dimensional output space y^{pred} via a hyperbolic tangent function for loss calculation and prediction. The hyperbolic tangent function $\tanh(x)$ maps the output values into the range between -1 and 1,

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}. \tag{8}$$

We calculate the root-mean-square error (RMSE) between the ground-truth sentiment score and the predicted score in the same batch as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(y_i^{\text{true}} - y_i^{\text{pred}}\right)^2}{N}}, \tag{9}$$

where N is the batch size, and y_{true} the is ground-truth sentiment score. We adopt Adam optimizer to train the regression model by minimizing the RMSE loss.

4 Experiment

4.1 Datasets

We conduct experiments on the datasets from "Fine-grained Sentiment Analysis on Financial Microblogs and News", which is Task 5 of SemEval-2017 [8] and was constructed by the SSIX project [9], to validate the proposed method. Two subsets are involved:

1. **Microblog Messages** include:
   (a) *StockTwits Messages*[1] consist of microblog messages posted on StockTwits platform that focus and discuss on stock market events and assessments from investors and traders.
   (b) *Twitter Messages*[2] include tweets and posts containing company stock symbols (cashtags $).
2. **News Statements & Headlines** include financial textual content crawled from different online resources, such as Yahoo Finance,[3] which are identified by the company names or abbreviations.

For the microblog dataset, only a unique token of the original post is provided for each instance. We retrieve the textual posts using Twitter API and StockTwits API. A brief statistics of both datasets are shown in Table 2.

4.2 Baselines

In this work, our goal is to validate the feasibility and effectiveness of incorporating statistical knowledge in the regression task. Therefore, we compare the models with and without statistical features (TCoL) using popular semantic feature extractors:

TextCNN [21] applies one-dimensional convolutional operation on the word embedding matrix and extracts latent feature vector by the max-over-time pooling.

BiLSTM [14] and **BiGRU** [3] are bi-directional models extracting both forward and reverse sequential features using LSTM and GRU cells.

We also compare with Twitter lexicon-based method [23] and traditional machine learning approaches, such as support vector machine (SVM) on the embedding vectors, N-gram XGBoost Regressor (XGBoost), and Multi-layer Perceptron (MLP) on the everaged embedding vector, which are also commonly used lightweight text regression models. Furthermore, we compare with top-ranked solutions [13, 20] in the SemEval challenge, which are ensemble-based approaches.

[1] https://stocktwits.com/

[2] https://twitter.com/

[3] http://finance.yahoo.com/

Table 2 Statistics of both training and test datasets of two tasks

Task	Domain	Set	Instance	Pos	Neg	Neutral
Microblog	StockTwits	Train	765*	246	510	9
		Test	371	116	243	6
	Twitter	Train	934*	330	586	18
		Test	429	141	280	8
Headline	Financial News	Train	1156	658	460	38
		Test	491	276	203	12

Pos and *Neg* stand for the numbers of instances with a positive sentiment score or a negative score. *Neural* stand for the number of corresponding instances whose sentiment score is zero

*The actual number of instances we used for training is far less than the reported statistics because of the damaged online resources

4.3 Word embedding and preprocessing

We adopt the publicly available pre-trained language model *FastText*[4] [35] as the word embedding model, which has 1 million word vectors with the dimensionality of 300. Words not present in the pre-trained model are initialized randomly (in this work, we adopt non-subword embedding instead of subword embedding, as using subword embedding produces lower results).

The mainstream preprocessing methods are following Kim [21], which removes most of signs as they have minor effects in the general domain. However, as discussed in Section 1, these signs play a non-trivial role in FSA. To examine the function of signs' statistics, we retain the signs that have special meanings in FSA, for example, $+$, $-$, $>$, $<$, %, and \$.

4.4 Evaluation metrics

To evaluate the model performance of both tasks, we calculate the **cosine similarity** between the ground-truth sentiment scores and the predicted scores. As the sentiment scores in FSA lie on a continuous scale between -1 and 1, cosine similarity compares the degree of agreement between gold standard and the predicted results. Given the vector of gold standard scores, G, and the vector of scores predicted by the model, P, the cosine similarity score is calculated as

$$cosine(G, P) = \frac{\sum_{i=1}^{n} G_i \times P_i}{\sqrt{\sum_{i=1}^{n} G_i^2} \times \sqrt{\sum_{i=1}^{n} P_i^2}}. \tag{10}$$

We compare the regression performance by reporting the root-mean-square error (RMSE). We also evaluate the model performance in a classification task, i.e., predicting the binary labels. We compare the *Macro*-average **F1 score** and the **Accuracy** score. The average results are reported based on ten trials for each model.

[4] https://fasttext.cc/docs/en/english-vectors.html

4.5 Parameter settings

The parameters in the CNN and RNN modules follow the same settings: the CNN-based models have filter size of [3, 4, 5] with 100 filters of each, and the RNN-based models have hidden dimension of 128. All models adopt Adam optimizer with batch size of 64 for microblog dataset and 32 for headline dataset. The dropout rate is set as 0.5. For the proposed methods, the dimension of encoded TCoL representation is 100.

4.6 Results

The results of our proposed method against other baseline methods are listed in Tables 3 (Task 1: Financial Microblogs) and 4 (Task 2: Financial Headlines). In Task 1, we compare the model performance with two different preprocessing methods. More concretely, we investigate the function of signs like $+$, $-$, $>$, $<$, and $\%$ by comparing models w/o signs and w/ signs, where signs are excluded or included, respectively. As for Task 2, we only test the models with signs removed as news headlines contain few signs. In general, our proposed method can achieve the best results on both tasks and yield consistent improvements to all baseline models, which validate the feasibility and the effectiveness of incorporating statistical information in FSA. Meanwhile, performance differences between models with and without signs considered are observed in the sentiment score prediction on financial microblogs, indicating the non-trivial function of signs in FSA.

5 Discussion

5.1 Performance differences on two tasks

From the overall results of the experiment, we can find that although there are significant improvements in both two tasks, the improvements in Task 1 are more evident than those in Task 2. For example, the highest growth of the cosine similarity score in Task 2 is 0.027, lower than all the improvements in Task 1. Meanwhile, our best result on Task 1 outperforms the second-best team according to the official release, while the best result on Task 2 can only achieve the Top 10. The differences indicate that our proposed model works better on the Financial Microblogs than Headlines.

The reason might be that statistical information such as positive or negative signs and the numbers takes a larger part in the financial microblogs. The format of a positive or negative sign plus a number or a percentage like $\$JD - 21.9\%$; $\$MS - 16.9\%$; $\$DAL - 15.0\%$ is easy to find in the financial microblogs. It conveys a clear message that the stock price is experiencing a rise or a drop and obvious sentiment information accordingly. The high frequency of the statistical information in the financial microblogs makes it discriminative, and thus improves the model performance marvelously. However, such a format seldom occurs in the financial headlines. The plain text or mix of different words with sentiments may dilute the single word's attribution to the sentence's TCoL matrix. Thus our proposed model struggles to identify whether the specific word is discriminative or not. Meanwhile, we observe that the entity name of persons like *Elon Musk* and *Tim Cook*, stock markets like *FTSE*, and companies like *Goldman Sachs*, *AstraZeneca*, and *Barclays* take a significant

Table 3 Results on Task 1: Financial Microblogs

Models	RMSE ↓	Accuracy (%) ↑	F1 Score (%) ↑	Cosine ↑
SVM	–	78.1	–	0.615
MLP	–	78.1	–	0.628
XGBoost	–	78.6	–	0.659
Lexicons	–	74.6	–	0.557
IITP [13]	–	–	–	**0.751**
RiTUAL [20]	–	–	–	0.70
CNN				
w/o signs	0.081	82.26	79.22	0.702
w/ signs	0.081	82.87	79.62	0.704
CNN + TCoL (Ours)				
w/o signs	0.071	84.20	81.30	0.748
Improv.	0.010	1.94%	2.08%	0.046
w/ signs	**0.070**	**85.03**	**82.60**	**0.751**
Improv.	0.011	2.16%	2.98%	0.047
BiLSTM				
w/o signs	0.089	80.07	77.19	0.671
w/ signs	0.085	81.10	78.13	0.684
BiLSTM + TCoL (Ours)				
w/o signs	0.085	81.48	78.91	0.698
Improv.	0.004	1.41%	1.72%	0.027
w/ signs	**0.080**	**82.60**	**79.36**	**0.717**
Improv.	0.005	1.50%	1.23%	0.033
BiGRU				
w/o signs	0.096	79.09	75.83	0.651
w/ signs	0.094	79.88	77.05	0.654
BiGRU + TCoL (Ours)				
w/o signs	0.086	80.76	77.69	0.692
Improv.	0.010	1.67%	1.86%	0.041
w/ signs	**0.084**	**81.60**	**78.26**	**0.700**
Improv.	0.010	1.72%	1.21%	0.046

↓ means the lower the better, and ↑ means the higher the better

proportion of the textual data, which compromises the effect of indicative words' statistics. Furthermore, due to the short news headlines, the data sparsity issue leads to low trustworthiness of the statistical knowledge as it may deviate from the real distribution. A promising solution is to exploit the Adaptive Gate model [25], which adjusts the information flow into the model by a Valve component, to filter out less necessary additional features and enhance model robustness.

Table 4 Results on Task 2: Financial News Headlines

Models	RMSE ↓	Accuracy (%) ↑	F1 Score (%) ↑	Cosine ↑
CNN	0.101	76.41	77.46	0.653
CNN + TCoL (Ours)	**0.092**	**78.58**	**78.90**	**0.680**
Improv.	0.009	2.17%	1.44%	0.027
BiLSTM	0.098	76.88	76.94	0.661
BiLSTM + TCoL (Ours)	**0.093**	**78.35**	**77.43**	**0.679**
Improv.	0.005	1.47%	0.49%	0.018
BiGRU	0.095	78.49	77.81	0.667
BiGRU + TCoL (Ours)	**0.092**	**79.56**	**79.43**	**0.680**
Improv.	0.003	1.07%	1.62%	0.013

↓ means the lower the better, and ↑ means the higher the better

5.2 Dimension of encoded TCoL vector

In this section, we discuss the effect of latent TCoL vector $\boldsymbol{\zeta}^z$'s dimension on the overall model performance. The dimension of the TCoL vector characterizes the scale of the latent space that generates word occurrences over labels. Variational encoding models with different latent dimensions present varied approximation performances. Therefore, the dimension of $\boldsymbol{\zeta}^z$ is a hyperparameter that has a potential influence on the model performance. We conducted additional experiments by encoding the TCoL matrix into latent spaces with different dimensions to examine such influence. The results with different semantic feature extractors are visualized in Figures 4 (with BiGRU) and 5 (with TextCNN). We observe that both models with BiGRU and TextCNN as the semantic feature extractor show apparent performance variety as the TCoL dimension changes. More concretely, both models yield poor results on all the metrics when the dimension is relatively small. The performances

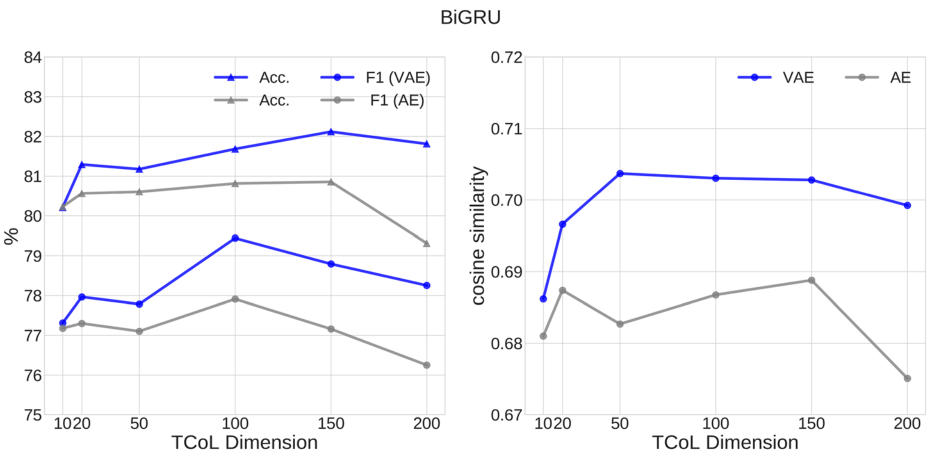

Figure 4 Effect of TCoL Vector $\boldsymbol{\zeta}^z$'s dimensionality to the model performance on Financial Microblog dataset with a BiGRU as semantic feature extractor

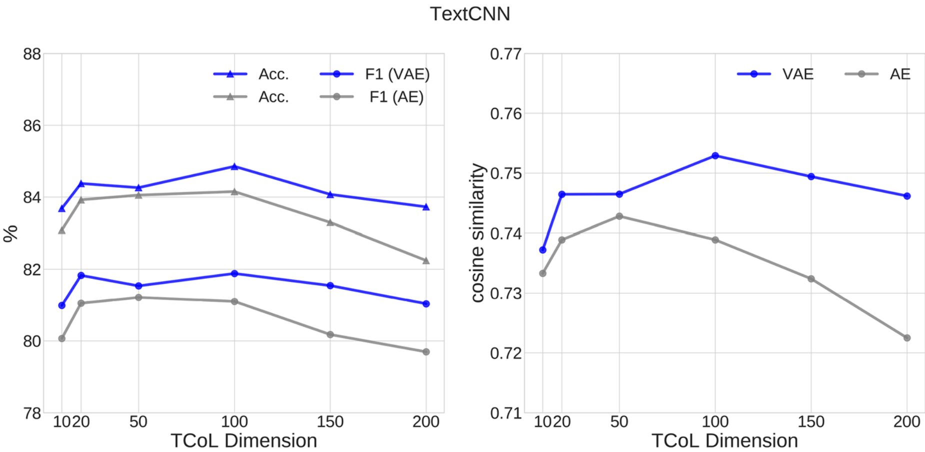

Figure 5 Effect of TCoL Vector ζ^z's dimensionality to the model performance on Financial Microblog dataset with a TextCNN as semantic feature extractor

gradually elevate as the dimension gets and peak at 100. Afterward, the performances deteriorate significantly when the dimension is greater than 200. As a result, we select 100 as the hyperparameter for latent TCoL's dimension in the experiment section of this work.

5.3 Autoencoder versus variational Autoencoder

In this section, we compare the performance of using Autoencoder (AE) and Variational Autoencoder to encode TCoL as they are both powerful representation learning models. In Figures 4 and 5, models' performances with a vanilla AE as the encoding module are also presented. From the visualizations, the VAE-based regression models consistently outperform the AE-based models, and the improvements are evident. Especially, the AE-based models show a severer performance deduction than VAE-based models when the dimension is greater than 200. We assume that this is because the probabilistic encoder in VAE restricts the latent space with a prior distribution, making the values in the learned representation more scalable than those encoded by an AE. Resembling the batch normalization technique, a scalable representation is beneficial to the neural decision-making process.

5.4 Scalability of implementation

In this section, we briefly analyze the computational cost of our proposed model to show that our method is scalable even with a large-scale dataset. The TCoL matrix with term's occurrence is generated at the preprocessing stage, which only needs to iterate the training set once with linear time complexity $\mathcal{O}(n)$. In the regression model, the encoded representation is directly concatenated with the semantic feature vector; thus, the additional cost on the model training is neglectable.

5.5 Case study

The function of signs towards model performance has been shown in experiments. We pick up 22 representative terms which are widely used in the financial domain from the sample texts. Table 5 shows their distributions on different categories. The words such as *Red* and *Green* usually describe the performance of the stock in the market. In the financial industry, *Red* always means an alert and a drop in the stock price, while *Green* has the opposite meaning. Thus, it is reasonable that 75% of sentences including *Red*, such as *5 Toxic Stocks Raising Red Flags to Sell*, are recognized as Negative or Very Negative, while 90% of sentences including *Green*, such as *$SPY wouldn't be surprised to see a green close*, are recognized as Positive or Very Positive.

Other terms such as the mathematical signs, i.e. $+$, $-$, $<$ and $>$, are common in the financial texts. All sentences including more than sign or less than sign are regarded as positive or negative respectively. The sentences with positive sign have about 90% probability to be classified to the Positive and Very Positive categories. Compared to the positive sign, the negative sign distributes randomly. The reason might be the negative sign not only represents a negative number but also has function as a short line between two single words. For example, the sentence *$GOOG and $MSFT -5.5%* means Google and Microsoft are experiencing a 5.5% drop, which is a negative sentiment information, while the sentence *$HP soars pre-market on business split announcement*, including a negative sign as well, has no pessimistic emotion.

In general, the statistic information extracted from the sample texts is intuitive and informative feature that addresses the tasks efficiently.

Table 5 Case Study

Terms	Very Neg.	Neg.	Neutral	Pos.	Very Pos.
Red	.125	.625	.0	.0	.250
Green	.0	.0	.100	.700	.200
Call	.0	.0	.0	.885	.115
Short	.132	.660	.113	.057	.038
Buy	.019	.074	.111	.389	.407
Downgrades	.0	1.0	.0	.0	.0
Losers	.0	1.0	.0	.0	.0
Losses	.333	.667	.0	.0	.0
Down	.0	.574	.131	.197	.098
Fall	.0	1.0	.0	.0	.0
Positive	.0	.0	.059	.588	.353
Negative	.0	1.0	.0	.0	.0
Holding	.0	.046	.136	.500	.318
Growth	.0	.059	.059	.529	.353
Rise	.0	.0	.667	.167	.166
Increase	.0	.0	.0	.500	.500
Bearish	.334	.333	.333	.0	.0
Bullish	.0	.0	.111	.083	.806
$+$	.0	.109	.328	.188	.375
$-$	.013	.316	.089	.291	.291
$>$	.0	.0	.0	.1.0	.0
$<$	.500	.500	.0	.0	.0

6 Conclusion and future work

This paper proposed an effective method to leverage corpus-level statistical information for the fine-grained financial sentiment analysis task. We partitioned the continuous range of sentiment score into five intervals and incorporated each term's occurrence on the intervals as a statistical feature to enhance the regression performance. We have conducted extensive experiments with CNN-based and RNN-based regression models to validate the feasibility and effectiveness of the proposed methods. The relationship between the terms and the labels is surprisingly informative in a domain-specific task. Compared with handcrafted lexicon resources, such statistical knowledge is easy to retrieve and flexible in application. Furthermore, we recognize the importance of signs in the financial sentiment analysis task, whose statistics are beneficial to the overall model performance.

For future work, we focus on the following points:

1. To refine the sentiment score intervals according to the frequency. We observe peaks and valleys in Figures 3 and 2, which means the sentiment scores are not evenly distributed due to possible human prior in annotating process or the *de facto* real-world distribution. This work adopts an empirical approach to split the score range to avoid introducing additional uncertainties. We will further investigate the influence of sentiment score distribution in our future research.
2. To investigate an effective method to understand numbers. Financial texts contain a large number of digits and numbers. It is important to understand the meaning of each number in the text. In this work, we only examine the signs' role in FSA, while the numbers, which determine the absolute value of a sentiment score, are not thoroughly discussed.
3. To incorporate additional domain knowledge together with the statistical features. Encoding statistical representation with VAE cannot fully prevent introducing noise to the model caused by the data sparsity issue. It would further improve the model performance by merging lexicon knowledge and statistical knowledge.

Acknowledgements The work of this paper has been supported by the Hong Kong Research Grants Council under the General Research Fund (Project No. 15200021), the Lam Woo Research Fund (Project No. LWI20011) and the Faculty Research Grant (Project No. DB21B6) of Lingnan University, Hong Kong, the One-off Special Fund from Central and Faculty Fund in Support of Research from 2019/20 to 2021/22 (Project No. MIT02/19-20) and the Research Cluster Fund (Project No. RG 78/2019-2020R) of The Education University of Hong Kong. Lau's work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11507219). Dian Zhang's work was supported by NSFC (Project No. 61872247).

References

1. Akhtar, M.S., Kumar, A., Ghosal, D., Ekbal, A., Bhattacharyya, P.: A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 540-546. Association for Computational Linguistics, Copenhagen, Denmark (2017). https://doi.org/10.18653/v1/D17-1057
2. Antweiler, W., Frank, M.Z.: Is all that talk just noise? the information content of internet stock message boards. J. Financ. **59**(3), 1259–1294 (2004). http://www.jstor.org/stable/3694736
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd International Conference on Learning Representations (2015)
4. Brown, G.W., Cliff, M.T.: Investor sentiment and the near-term stock market. J. Empir. Financ. **11**(1), 1–27 (2004). https://doi.org/10.1016/j.jempfin.2002.12.001, https://www.sciencedirect.com/science/article/pii/S0927539803000422

5. Cai, Y., Huang, Q., Lin, Z., Xu, J., Chen, Z., Li, Q.: Recurrent neural network with pooling operation and attention mechanism for sentiment analysis: A multi-task learning approach. Knowledge-Based Systems **203**, 105856 (2020)
6. Cambria, E., Li, Y., Xing, F.Z., Poria, S., Kwok, K.: Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 105–114 (2020). Association for Computing Machinery
7. Chen, X., Xie, H., Cheng, G., Li, Z.: A decade of sentic computing: Topic modeling and bibliometric analysis. Cogn. Comput., 1–24 (2021)
8. Cortis, K., Freitas, A., Daudert, T., Huerlimann, M., Zarrouk, M., Handschuh, S., Davis, B.: SemEval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 519–535. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/S17-2089
9. Davis, B., Cortis, K., Vasiliu, L., Koumpis, A., McDermott, R., Handschuh, S.: Social sentiment indices powered by x-scores. 2nd International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA 2016. p. 21 (2016)
10. Do, H.H., Prasad, P., Maag, A., Alsadoon, A.: Deep learning for aspect-based sentiment analysis: A comparative review. Expert Syst. Appl. **118**, 272–299 (2019). https://doi.org/10.1016/j.eswa.2018.10.003, https://www.sciencedirect.com/science/article/pii/S0957417418306456
11. Fama, E.F.: Efficient capital markets: A review of theory and empirical work. J. Financ. **25**(2), 383–417 (1970). http://www.jstor.org/stable/2325486
12. Feuerriegel, S., Prendinger, H.: News-based trading strategies. Decis. Support Syst. **90**, 65–74 (2016)
13. Ghosal, D., Bhatnagar, S., Akhtar, M.S., Ekbal, A., Bhattacharyya, P.: IITP at SemEval-2017 task 5: An ensemble of deep learning and feature based models for financial sentiment analysis. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 899–903. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/S17-2154
14. Graves, A., Jaitly, N., Mohamed, A.: Hybrid speech recognition with deep bidirectional lstm. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 273–278 (2013). https://doi.org/10.1109/ASRU.2013.6707742
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
16. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 168–177. Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1014052.1014073
17. Hutto, C., Gilbert, E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 8 (2014)
18. Jiang, M., Lan, M., Wu, Y.: ECNU at SemEval-2017 task 5: An ensemble of regression algorithms with effective features for fine-grained sentiment analysis in financial domain. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 888–893. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/S17-2152
19. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 562–570. Association for Computational Linguistics (2017)
20. Kar, S., Maharjan, S., Solorio, T.: RiTUAL-UH at SemEval-2017 task 5: Sentiment analysis on financial data using neural networks. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 877–882. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/S17-2150
21. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751. Association for Computational Linguistics (2014)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the 2014 International Conference on Learning Representations (2014)
23. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. J. Artif. Intell. Res. **50**, 723–762 (2014)
24. Li, F.: The information content of forward-looking statements in corporate filings—a naïve bayesian machine learning approach. J. Account. Res. **48**(5), 1049–1102 (2010). https://doi.org/10.1111/j.1475-679X.2010.00382.x
25. Li, X., Li, Z., Xie, H., Li, Q.: Merging statistical feature via adaptive gate for improved text classification. Proc. AAAI Conf. Artif. Intell. **35**(15), 13288–13296 (2021). https://ojs.aaai.org/index.php/AAAI/article/view/17569
26. Li, X., Li, Z., Zhao, Y., Xie, H., Li, Q.: Incorporating effective global information via adaptive gate attention for text classification. arXiv:2002.09673 (2020)

27. Li, X., Xie, H., Chen, L., Wang, J., Deng, X.: News impact on stock price return via sentiment analysis. Knowl.-Based Syst. **69**, 14–23 (2014). https://doi.org/10.1016/j.knosys.2014.04.022, https://www.sciencedirect.com/science/article/pii/S0950705114001440
28. Li, X., Xie, H., Lau, R.Y.K., Wong, T., Wang, F.L.: Stock prediction via sentimental transfer learning. IEEE Access **6**, 73110–73118 (2018)
29. Li, Z., Chen, X., Xie, H., Li, Q., Tao, X.: Emochannelattn: Exploring emotional construction towards multi-class emotion classification. In: 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), pp. 242–249 (2020). https://doi.org/10.1109/WIIAT50758.2020.00036
30. Li, Z., Xie, H., Cheng, G., Li, Q.: Word-level emotion distribution with two schemas for short text emotion classification. Knowl.-Based Syst., 107163. https://doi.org/10.1016/j.knosys.2021.107163, https://www.sciencedirect.com/science/article/pii/S0950705121004263 (2021)
31. Loughran, T., Mcdonald, B.: When is a liability not a liability? textual analysis, dictionaries, and 10-ks. J. Financ. **66**(1), 35–65 (2011). https://doi.org/10.1111/j.1540-6261.2010.01625.x
32. Luo, L., Ao, X., Pan, F., Wang, J., Zhao, T., Yu, N., He, Q.: Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pp. 4244–4250 (2018). https://doi.org/10.24963/ijcai.2018/590
33. Mai, L., Le, B.: Joint sentence and aspect-level sentiment analysis of product comments. Ann. Oper. Res., 1–21 (2020)
34. Malkiel, B.G.: The efficient market hypothesis and its critics. J. Econ. Perspect. **17**(1), 59–82 (2003). https://doi.org/10.1257/089533003321164958
35. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp. 52–55. European Language Resources Association (ELRA) (2018). https://www.aclweb.org/anthology/L18-1008
36. Mohammad, S.M., Kiritchenko, S.: Using hashtags to capture fine emotion categories from tweets. Comput. Intell. **31**(2), 301–326 (2015). https://doi.org/10.1111/coin.12024
37. Mohammad, S.M., Turney, P.D.: Nrc emotion lexicon. National Research Council, Canada, pp. 1–234 (2013)
38. Mowlaei, M.E., Saniee Abadeh, M., Keshavarz, H.: Aspect-based sentiment analysis using adaptive aspect-based lexicons. Expert Syst. Appl. **148**, 113234 (2020). https://doi.org/10.1016/j.eswa.2020.113234, https://www.sciencedirect.com/science/article/pii/S0957417420300609
39. Ramos, J. et al.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning (2003)
40. Sabherwal, S., Sarkar, S.K., Zhang, Y.: Do internet stock message boards influence trading? evidence from heavily discussed stocks with no fundamental news. J. Bus. Financ. Account. **38**(9-10), 1209–1237 (2011). https://doi.org/10.1111/j.1468-5957.2011.02258.x
41. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
42. Stone, P.J., Dunphy, D.C., Smith, M.S.: The general inquirer: A computer approach to content analysis (1966)
43. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)
44. Wang, G., Wang, T., Wang, B., Sambasivan, D., Zhang, Z., Zheng, H., Zhao, B.Y.: Crowds on wall street: Extracting value from collaborative investing platforms. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work &; Social Computing, CSCW '15, pp. 17–30. Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2675133.2675144
45. Wang, Q., Lau, R.Y.K.: The impact of investors' surprise emotion on post-m&a performance: A social media analytics approach. In: 40th International Conference on Information Systems (ICIS 2019) (2019). Association for Information Systems
46. Xing, F., Malandri, L., Zhang, Y., Cambria, E.: Financial sentiment analysis: An investigation into common mistakes and silver bullets. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 978–987. International Committee on Computational Linguistics, Barcelona Spain (Online) (2020). https://doi.org/10.18653/v1/2020.coling-main.85
47. Xing, F.Z., Cambria, E., Welsch, R.E.: Natural language based financial forecasting: a survey. Artif. Intell. Rev. **50**(1), 49–73 (2018)
48. Xu, J., Cai, Y., Wu, X., Lei, X., Huang, Q., Leung, H.F., Li, Q.: Incorporating context-relevant concepts into convolutional neural networks for short text classification. Neurocomputing **386**, 42–53 (2020)

49. Yuan, H., Tang, Y., Xu, W., Lau, R.Y.K.: Exploring the influence of multimodal social media data on stock performance: an empirical perspective and analysis. Internet Res. (2021)
50. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: Proceedings of Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, pp. 649–657 (2015)
51. Zubiaga, A.: Exploiting Class Labels to Boost Performance on Embedding-Based Text Classification. Association for Computing Machinery, New York, NY USA. https://doi.org/10.1145/3340531.3417444 (2020)

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

Affiliations

Han Zhang[1] · Zongxi Li[2] · Haoran Xie[3] · Raymond Y. K. Lau[4] · Gary Cheng[5] · Qing Li[6] · Dian Zhang[7]

Han Zhang
haileyzhang@cuhk.edu.hk

Haoran Xie
hrxie2@gmail.com

Raymond Y. K. Lau
raylau@cityu.edu.hk

Gary Cheng
chengks@eduhk.hk

Qing Li
csqli@comp.polyu.edu.hk

Dian Zhang
zhangd@szu.edu.hk

[1] Department of Finance, The Chinese University of Hong Kong, Shatin, Hong Kong

[2] School of Science and Technology, Hong Kong Metropolitan University, Ho Man Tin, Hong Kong

[3] Department of Computing and Decision Sciences, Lingnan University, Tuen Mun, Hong Kong

[4] Department of Information Systems, City University of Hong Kong, Kowloon, Hong Kong

[5] Department of Mathematics and Information Technology, The Education University of Hong Kong, Tai Po, Hong Kong

[6] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[7] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China