Neural Networks 164 (2023) 115–123

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2022 Special Issue

Explainable hybrid word representations for sentiment analysis of financial news

Surabhi Adhikari [a], Surendrabikram Thapa [b], Usman Naseem [c], Hai Ya Lu [d], Gnana Bharathy [d], Mukesh Prasad [d,*]

[a] Department of Computer Science and Engineering, Delhi Technological University, New Delhi, India
[b] Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA
[c] School of Computer Science, University of Sydney, Sydney, Australia
[d] School of Computer Science, University of Technology Sydney, Sydney, Australia

ARTICLE INFO

Article history:
Available online 21 April 2023

Keywords:
Hybrid word embeddings
XAI
Explainable sentiment analysis
Natural Language Processing
Contextual embeddings
Explainability

ABSTRACT

Due to the increasing interest of people in the stock and financial market, the sentiment analysis of news and texts related to the sector is of utmost importance. This helps the potential investors in deciding what company to invest in and what are their long-term benefits. However, it is challenging to analyze the sentiments of texts related to the financial domain, given the enormous amount of information available. The existing approaches are unable to capture complex attributes of language such as word usage, including semantics and syntax throughout the context, and polysemy in the context. Further, these approaches failed to interpret the models' predictability, which is obscure to humans. Models' interpretability to justify the predictions has remained largely unexplored and has become important to engender users' trust in the predictions by providing insight into the model prediction. Accordingly, in this paper, we present an explainable hybrid word representation that first augments the data to address the class imbalance issue and then integrates three embeddings to involve polysemy in context, semantics, and syntax in a context. We then fed our proposed word representation to a convolutional neural network (CNN) with attention to capture the sentiment. The experimental results show that our model outperforms several baselines of both classic classifiers and combinations of various word embedding models in the sentiment analysis of financial news. The experimental results also show that the proposed model outperforms several baselines of word embeddings and contextual embeddings when they are separately fed to a neural network model. Further, we show the explainability of the proposed method by presenting the visualization results to explain the reason for a prediction in the sentiment analysis of financial news.

1. Introduction

Finance has been an indispensable part of human life since the origin of human civilization. It is undoubtedly prominent from the earliest trend of barter systems to today's advanced cryptocurrencies (Gupta, Dengre, Kheruwala, & Shah, 2020). Finance has broadly been related to data, specifically transactions, prices, stocks, reports, and accounts. It is noteworthy that people, especially in today's digital apogee, have significantly inclined towards investment, and the share market tempts many. With the unprecedented growth of the Internet, investors worldwide have easy access to opportunities to gather and share their experiences. Websites like StockTwits, SeekingAlpha, etc., and microblogging sites like Reddit, Twitter, etc., have mainly served this purpose (Sohangir, Wang, Pomeranets, & Khoshgoftaar, 2018). Through the Internet, people are quickly gaining access to get advice from investment experts. Moreover, the stock market, being an open market, reflects the goods and resources exchanged in an economy. With a slight change in the economy caused by any governmental transformations or private sectors' advancement in the economic competition, all the economy participants reequip their states in the market, and the prices that directly determine their values, adjust accordingly (Araci, 2019). Financial media, which mainly comprises financial news, posts of the competitors on social media, analysts' predictions, collectively help investors feed in the information. Through these media, the investors develop a perspective and intuition to invest in a particular company. Sentiment analysis has played a significant role in forming a prescience among investors regarding a company's future values. Stock market prediction is one of the essential applications in which sentiment analysis has been extensively used

* Corresponding author.
E-mail address: mukesh.nctu@gmail.com (M. Prasad).

https://doi.org/10.1016/j.neunet.2023.04.011
0893-6080/

to predict future stock market trends and prices from financial texts' exegesis.

Due to the current development in deep learning and NLP techniques, there has been much work in the automatic analysis of sentiments of people over the internet. Sentiment analysis, also widely known as opinion mining (Adhikari, Thapa, Singh, Huo, Bharathy, & Prasad, 2021), is extensively used in many domains, such as microblogs, e-commerce sites, and social media. The primary objective of sentiment analysis can be widely categorized as emotion recognition and polarity identification. Emotion detection is more inclined towards mining a set of emotion labels; on the other hand, polarity detection is a category-oriented method with discrete outputs such as positive or negative. Initially, lexicon-based approaches were widely used for sentiment analysis. In this method, the entire text's overall score is calculated from the number of positive and negative words present in the text. Similarly, classical machine learning approaches have also been widely adopted for sentiment analysis. With the enormous success and prospects of deep learning, sentiment analysis has been vigorously explored using deep learning techniques in recent years (Chowdhury, Sil, & Shukla, 2021; Sun & Chu, 2020).

The applications of natural language processing in the financial domain comprise analyzing financial texts that include asset allocation, credit scoring, stock market, initial public offering (IPO), market/foreign exchange, and more (Bai, Xing, Cambria, & Huang, 2019; Day & Lee, 2016; Xing, Cambria, Malandri, & Vercellis, 2019). Applying the NLP techniques mainly includes two methods to process the financial textual inputs (Soares Koshiyama, Firoozye, & Treleaven, 2020). The first method is to straightaway encode the financial texts with the help of neural networks to use the learning representations for down streaming tasks (Xu, Qiu, Zhou, & Huang, 2020). The other method is to delineate the critical linguistic features such as the semantics of the content (Sheng et al., 2022; Yang, Zhu, Wang, Wang, & Z, 2022), stakeholders' sentiments, and investors' sentiments (Guo et al., 2022; Huang, Han, Li, Wang, & W, 2021; Meng, Xiao, & Wang, 2022; Xiong, Weng, & Y, 2022). Financial sentiment analysis (FSA) is one of the most common applications of NLP using the second method. The major objective of the FSA is to classify a given piece of financial text if it depicts bullish or bearish expressions towards certain arguments. These arguments could be a change in law, merging of financial institutions, or opening of an IPO. FSA stands to be a challenging task because of the want for large-scale training data. The other major challenge is the difficulty in annotating/labeling after collecting the texts, requiring scrutiny and expert knowledge. Therefore, the models for FSA usually perform worse than the same sentiment analysis models for the general domain.

Language models and various word representations are being used to aid the deep learning models in understanding text data and inferring useful information (Thapa et al., 2020). Through learning from the different corpus using sophisticated algorithms, the language models and word representations can depict the meaning of the word precisely. Despite the high accuracy of word representations, they cannot distinguish the contextual meanings of the words. In the English language, a single word can have multiple meanings. The meanings are not necessarily the same for words and can differ according to sentences and context. For example, the same word used in the medical field might have entirely different meanings in the financial domain. Thus, we need representation that can capture such polysemy (Naseem & Musial, 2019). Our research should ensure that learned representations: (i) capture polysemy in the context and (ii) represent complicated attributes of words used, including semantics and syntax. Further in our paper, we show how models' prediction abilities vary with hybrid embeddings and how they perform with standard word representations.

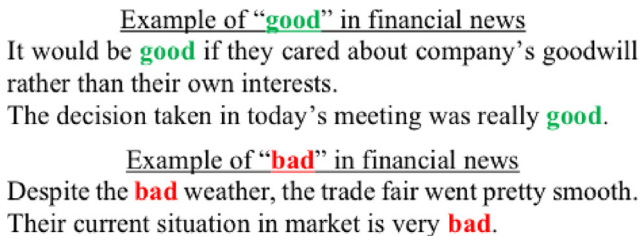

Fig. 1. Words with different meanings in different contexts.

For example, in Fig. 1 shown above, the word "good" and "bad" are used in both positive and negative contexts. Thus, the word representations must be able to catch what they mean in the given sentences. The representations can be rich in features using a hybrid representation model based on more than one pre-existing embedding technique (Naseem, Razzak, Musial, & Imran, 2020). Also, incorporating linguistic features such as Part of Speech (POS) in word representations can be phenomenal (Manning, 2011). Thus, in this paper, we propose a novel deep learning-based architecture based on hybrid word representations that best classify sentiments of financial texts. Financial Phrasebank Dataset (Malo, Sinha, Korhonen, Wallenius, & Takala, 2014) has been utilized in the study. The significant contributions of the paper are:

- A novel explainable hybrid architecture combining static and contextual word representations has been developed which addresses language ambiguity and is devised to comprehensively capture polysemy in context, semantics, and syntactical information of words.
- Due to the imbalanced class in the dataset, we have augmented the dataset to bring uniformity in the class that eventually performs better.
- Extensive experiments are conducted on several real-world datasets to evaluate the proposed method. All the results prove that our model constantly outperforms other state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 reviews the literature on the topic. Section 3 describes the model and methodology opted in the experiments. Section 4 gives an in-depth analysis of the results obtained along with an explainability of the models and embeddings and Section 5 concludes the paper with challenges and future scopes of improvements.

2. Related works

Sentiment analysis is a domain of Natural Language Processing that is chiefly used to analyze people's sentiments through texts. Initially, supervised learning with the tagged annotations is used to classify the text as positive or negative (Wiebe, Bruce, & O'Hara, 1999). Researchers have recently started exploring the field and incorporated the latest Natural Language Processing techniques to develop the best models (Parihar, Thapa, & Mishra, 2021). Renault et al. (Renault, 2020) scraped StockTwits and applied machine learning algorithms for sentiment analysis of financial texts. His work showed the best performances with linear SVM in his dataset. Loughran et al. (Arnold, 2011) have used the US Security and Exchange Commission portal from 1994 to 2008 to make a financial lexicon and manually create six-word lists including positive, negative, litigious, uncertainty, model strong, and model weak.

Similarly, supervised classification methods, such as Support Vector Machines (Krishnamoorthy, 2018), Naïve Bayes (Saif, He,

& Alani, 2012), or ensembles (da Silva, Hruschka, & Hruschka, 2014; Fersini, Messina, & Pozzi, 2014) have been deployed to perform sentiment analysis in multiple financial datasets on various research projects. Min-Yuh Day et al. (Day & Lee, 2016) analyzed the relationships between financial news and the trend of the stock price of that Day. They used different deep learning-based architectures to determine whether the stock markets will vary based on people's opinions on different sources and to what degree they influence the investors' decisions. Yeh, Yeh, and Shen (2020) proposed different methods for word vector representations for modeling financial information. They explored bag-of-words, domain-specific, and pre-trained word embeddings. Later, the embeddings were applied with linear and non-linear methods to form a text regression architecture for volatility prediction. Sehrawat (2019) published the word embeddings learned from 10-K filings. These word embeddings are proved significant for differentiating between various types of sentiment words on financial documents and could be used for tasks such as document similarity, sentiment analysis, readability index, etc. Word embeddings have given promising results, but a single embedding is not always perfect for representing words, as explained earlier (Naseem, Khan, Razzak, & Hameed, 2019).

Much research is being done to make word representations more robust and richer in features to tackle this problem. Naseem, Razzak, Eklund, and Musial (2020) proposed a deep contextual-based embedding for identifying ironical and sarcastic posts in social networks. The transformer-based contextual embeddings improved the noise within contexts and solved the ambiguities in languages like polysemy and word sentiments usually becomes too inclined towards the majority class. So, to tackle both polysemy and class imbalance, we propose a hybrid representation of words based on multiple embeddings for the identification of financial sentiments from the Financial Phrasebank dataset.

3. Exaplainaible hybrid word representations

The proposed explainable architecture for the financial sentiment analysis consists of the hybrid embeddings and CNN or BiLSTM or a Hybrid of CNN and BiLSTM with attention atop it. The complete block diagram of the proposed architecture is shown in Fig. 2. The embeddings for the financial phrase bank dataset are extracted using pre-trained word2vec, pre-trained BERT, and POS tags. Word2vec embeddings extract features from the text by capturing the word semantics information. Similarly, BERT embeddings capture the context and thus overcome the language ambiguity and polysemy problem. On the other hand, the POS vectors extract the tags of speech and are therefore useful for the text's syntactic information. The vectors are then concatenated to form a representation hybrid layer. These vectors are then fed to a neural network model. There is attention applied to the output of the last hidden layer for sentiment classification based on the sentiments. Each of the steps in the proposed architecture is described below in detail.

3.1. Data augmentation

It can be seen from Table 2 that there is a huge imbalance in the classes in the original dataset. There are significantly lower negative and positive sentiments than neutral sentiments. It is cumbersome to annotate a large amount of data, and this can lead to less amount of training data. Hence, it is extremely important to use the necessary augmentation techniques to get enough amount of data. Also, in NLP augmenting data can be a difficult task because of the grammatical structure of the language.

There are a few augmentation techniques such as oversampling and undersampling to increase the size of the data. The classical oversampling techniques like SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) do not work well with text data because of the linguistic complexities. Yu et al. (2018) proposed a mechanism to translate a sentence to French and again back to English for text data augmentation. Similarly, Kafle et al. (Ioffe & Szegedy, 2015) proposed a semantic annotation-based technique that generated new texts. Although these techniques are valid, it is still challenging to come up with a generalized and universal rule for augmenting language data given the complexity of the language. To overcome the problem of class imbalance, various text data augmentation techniques have been utilized. The augmentations were done using the nlpaug library in Python (Liu et al., 2019). Contextual synonyms based on BERT (Devlin, Chang, Lee, & Toutanova, 2018) and RoBERTa (Liu & Guo, 2019) and synonyms from the WordNet were used to substitute the words in the text. An example of a new sentence after giving the original sentence as input can be found in Table 1. The distribution of the data after augmentation can be found in Table 2.

3.2. Representation layer

Hybrid representation layer is the concatenation of three different vectors, namely, word2vec ($V_{Word2Vec}$), BERT (V_{BERT}), and POS (V_{POS}). Word2vec is essential for capturing the similarity between words in a textual corpus. Similarly, BERT represents contextual information and is hence a key to overcoming the polysemy problem. POS, on the other hand, is important for syntactical features and extracting the grammatical properties in a sentence. The details of each of the embeddings are described below.

3.2.1. Contextual embeddings

The ability to capture the context is extremely important for the representation of words (Devlin et al., 2018). This is essential for handling the polysemy problem. BERT was originally designed for representing deep bidirectional contextual texts by considering both the left and right contexts of a sentence. The positional embeddings of the BERT model capture the positions of words in a sentence thus making it more robust for understanding the context of words within a text. BERT vectors have a dimension of 768. The vectors were finally concatenated to the word2vec and POS embeddings to get the hybrid representation layer.

3.2.2. Word embeddings

The words in the sentences are assigned real-valued vectors using word embeddings. Word embeddings are based on the idea that if features have similar meanings, then it is useful to represent the features that depict this similarity. Bengio et al. (Thapa, Adhikari, & Mishra, 2021) proposed a probabilistic neural model where the vocabulary words were mapped to a distributed word feature vector. The feature vector represents several aspects of the word. These features are smaller than the size of the vocabulary. Pretrained word2vec (Mikolov, Chen, Corrado, & Dean, 2013) is used in the experiments to produce embeddings. The embeddings had the same size of 756. The maximum number of words in the sentences was 81. Hence, the input size was (81, 756). Similarly, the pretrained fastText (Grave, Bojanowski, Gupta, Joulin, & Mikolov, 2018) model of embedding size 768 was also used for the baseline models.

3.2.3. Part-of-speech (pos) embeddings

POS tagging is essential in NLP tasks because it assigns the appropriate POS tag to each word in the context. POS gives us useful information about a word, its neighbors, and different syntactic categories of words such as verbs, nouns, adverbs, adjectives, etc. Our proposed model has used the Stanford parser for

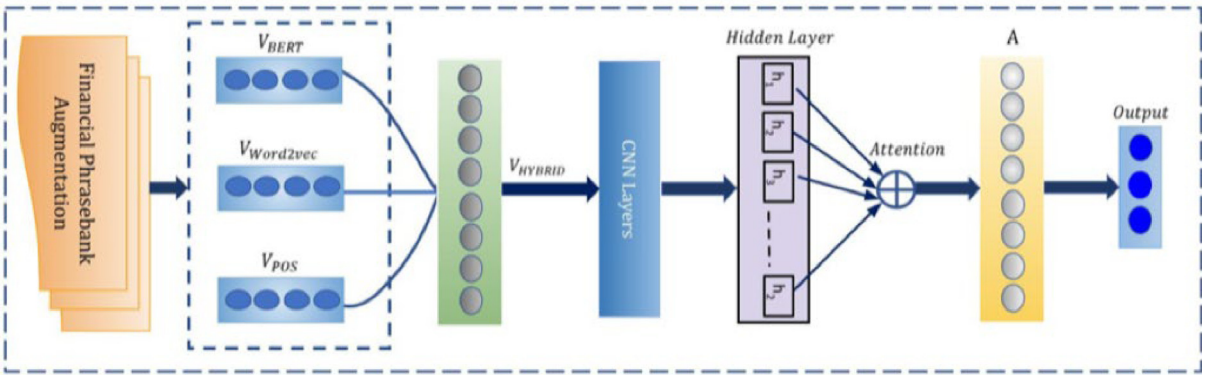

Fig. 2. Hybrid representation with neural network models and attention.

Table 1
Example of data augmentation.

Original Data	Augmented Data
Shares of Nokia Corp. rose Thursday after the cellphone maker said its third-quarter earnings almost doubled and its share of the global handset market increased.	Shares of Nokia Corp. increased after the cell phone manufacturer said its third-quarter earnings nearly doubled its share of the global handset market become larger.
The last quarter was the best quarter of 2009 in net sales, and the operating margin rose to 12.2%.	The final quarter was the best quarter of 2009 in net sales because the operating margin reached 12.2%.

POS tagging, which generates POS tags. Each POS-tagged token is then transformed into a vector, of dimension 12. The POS vectors (V_{POS}) were then concatenated to the initial word2vec embeddings, giving us a dimension (81, 768). These vectors were finally concatenated to get the hybrid representation layer:

$$V_{Hybrid} = V_{Word2Vec} + V_{BERT} + V_{POS} \quad (1)$$

Where,
V_{Hybrid} = Hybrid word embeddings
$V_{Word2Vec}$ = Word2Vec embeddings
V_{BERT} = BERT embeddings
V_{POS} = POS vectors

3.3. Classifiers

3.3.1. Convolutional neural network

We have used the famous Kim's CNN architecture (Kim, 2014) in this study. In Kim's architecture, after every convolutional layer, max pooling is applied. In this experiment, three convolutional layers with tanh as the activation function have been used, and after each layer, a max pool of filter size three has been applied. The flattened layer after the last max pool filter reshapes the input size, followed by the dropout layer with a rate of 0.5. Max pooling helps to avoid over-fitting by facilitating a model with an abstracted form of the hybrid representations. It also helps to reduce the computational cost and thereby makes the model useful in use cases where real-time decisions are to be made. The dropout layer randomly sets inputs to 0 and prevents overfitting. The dense layer with two nodes, also the output layer, has SoftMax as the activation function that transforms the results into probabilities of each class. The number of epochs and batch size have been fixed to 20 and 50 respectively. The CNN layer takes the input of the given vector and V_{Hybrid} gives the representation of h_{CNN} which is then fed to the attention layer.

3.3.2. Bi-directional Long Short-Term Memory (BiLSTM)

Unlike Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), in BiLSTMs, the signal propagates in both directions, i.e. backward and forward. BiLSTMs train first on the input sequence and then on the reversed input sequence. The forget, input, and output gates and the cell states decide what information to throw away, update the cells, and then produce the output by carrying only the relevant information. In this work, four BiLSTM cells with 16, 8, 4, and 2 nodes subsequently and tanh as the activation functions have been used. The input is the same as that of the convolutional layer, i.e. the vector of word embeddings. After the first BiLSTM layer, a dropout with a 0.5 rate has been used for regularization. After the three BiLSTM layers, again a dropout of 0.25 has been used. The output of the BiLSTM cell has been connected to a dense layer with four nodes and ReLU as the activation function. The output layer has softmax as the activation function in order to predict probabilities for the two categories. To prevent overfitting, L2 regularizers have been used. Adam optimizer has been used. The number of epochs and batch size has haven fixed to 20 and 50 respectively for all embeddings. Similarly, the BiLSTM layer also takes the input of V_{Hybrid} with the sequences of x_n tokens and produces the hidden representations of h_{BiLSTM} by concatenating the hidden representations from both forward ($\overrightarrow{h}_{LSTM}$) and backward ($\overleftarrow{h}_{LSTM}$) LSTM and is given by the Eq. (2).

$$h_{BiLSTM} = [\overrightarrow{h}_{LSTM} \parallel \overleftarrow{h}_{LSTM}] \quad (2)$$

Table 2
Distribution of data in Phrasebank.

Before Augmentation		After Augmentation	
All agree	Merged Data	All Agree	Merged Data
570	1363	1140	2726
303	604	1212	2416
1385	2872	1385	2872
2258	4839	3737	8014

3.3.3. CNN with BiLSTM cells

CNNs learn the local features of the text, and RNNs learn long-term dependencies. Combining these architectures can give better performance on various NLP tasks such as sentiment analysis and text classification (Zhao, Gao, Wen, & Li, 2021). In this experiment, four convolutional layers and two BiLSTM cells have been used. The word embeddings are fed to the convolutional layer. After

every two convolutional layers, a max-pooling of size three has been applied. To prevent overfitting, L2 regularizers have been used in both the networks. Tanh has been used as the activation function for the BiLSTM cells. After the first BiLSTM cell, batch normalization (Kadlec, Schmid, Bajgar, & Kleindienst, 2016) has been done. Adam optimizer has been used. The output of the BiLSTM cell has been connected to a fully dense layer with ReLu as the activation function and twenty nodes. One more dense layer has been added with ReLU as the activation function with ten nodes. The softmax function in the output layer transforms the vectors to predict the category of the transcripts. The number of epochs and batch size have been fixed to 20 and 50 respectively. Similarly, for hybrid of CNN and BiLSTM model, the hidden representations of $h_{CNN+BiLSTM}$ is produced.

3.4. Attention layer

Attention mechanisms are used in encoder–decoder architectures where it attends to the encoder and previous hidden states. With an input sentence and all the associated hidden states, attention layers decide what part of the input was most relevant and useful with each output instance. Attention preserves the context from beginning to end hence achieving great results on various NLP tasks such as machine translation (Choi, Cho, & Bengio, 2018), text summarization (Bengio, Schwenk, Senécal, Morin, & Gauvain, 2006), text classification (Li & Shah, 2017), etc. All the deep learning models used in this study have also been trained with an attention layer (Huang et al., 2022; Liu et al., 2022; Lu, Zhu, Yin, Yin, & L, 2022; Vaswani et al., 2017; Wang & Li, 2022). Apart from being able to attend to the encoder and previous hidden states, attention can also be used to get a distribution over features, such as the word embeddings of a text (Kafle, Yousefhussien, & Kanan, 2017). The attention used in this study is the multiplicative self-attention layer because of its space efficiency and less operation time. Self-attention (Salazar, Kirchhoff, & Huang, 2019) is used to extract the relevant features of a sentence by enabling it to attend to itself. The architecture of the models is the same as described above, with only an attention layer after the first layers in every model. In any sentence, there are words that play a greater role than other words. Thus, if a model drops non-important words, it can be expected to perform better. Thus, the attention mechanism was used to give weightage to relatively more important words. Attention mechanism, in short, works by assigning appropriate weightage to each token through a SoftMax function and the representation after attention, A is calculated as a weighted sum of all tokens and given by Eq. (3).

$$A = \sum_{j=1}^{n} a_j h_j \tag{3}$$

where,

$$a_j = \frac{\exp\left(e_j\right)}{\sum_{z=1}^{n}\left(\exp\left(e_z\right)\right)}; \quad \sum_{j=1}^{n} a_j = 1$$

$$e_j = \tanh(W_h h_j + b_h)$$

Where, W_h and b_h are the learned parameters; h_j is the representations from the models (CNN or BiLSTM or hybrid of CNN and BiLSTM).

3.5. Output layer

The final representation, A which is obtained after the attention mechanism is now fed to a fully connected SoftMax layer to obtain the class probability distribution. The categorical Cross-entropy loss function which is also our objective function was minimized. From Eq. (4), the loss function, L_{CCE} decreases as the predicted probability p_i converges towards the ground truth g_i.

$$L_{CCE} = -\sum_{i}^{C} g_i \log(\mathrm{f}\,(p_i)) \tag{4}$$

Where, g_i and p_i are ground truths and predicted probability respectively. L_{CCE} represents categorical cross-entropy loss.

4. Experimental results

The baseline is established with various deep-learning models. The features with Word2Vec, fastText, and BERT are used separately with these models. In this section, the dataset used in the experiment is described in detail along with the experimental results. Further, later in the section, the past works in the same domain have been discussed to draw a comparison between the proposed model and existing literature.

4.1. Dataset

The data used for this study is the Financial Phrase Bank (Malo et al., 2014). The dataset comprises financial news found on the LexisNexis database. There are datasets for different levels of annotator agreements. In this experiment, all agree, i.e., the dataset with 100% inter-annotator agreement, and merged data which is a combination of all the other datasets of different agreement levels given in the Financial Phrasebank has been used. The financial news is labeled as positive, neutral, and negative for news showing positive, neutral, and negative sentiments respectively. The distribution of each of the classes in all agreed data has been shown in Table 2. Further, after the augmentation, the class imbalance problem has been taken care of and the number of texts after augmentation has been shown in Table 2. Besides, the Financial Phrase Bank dataset has all annotations from a single organization. This imposes a limitation that organizational biases would be embedded in the annotation.

4.2. Performance measures

In all the above-mentioned architectures, validation has been done using stratified 10-fold cross-validation. The performance of the proposed architectures has been measured and compared with respect to the accuracies of each of the models. For each of the learning models, evaluation of the performance was done using accuracy and f1-score.

4.2.1. Baselines

Our model is compared with the baseline models discussed in Section 3. Word2vec, fasttext, and BERT embeddings were used to train the deep learning architectures. The baseline models use only a single word embedding i.e. either contextual word embedding or static word embedding. With the given data, the CNN model with word2vec embeddings has performed the best with an accuracy of 0.79 and an f1-score of 0.76. Likewise, for the merged dataset, the same CNN model has shown the best performance with an accuracy of 0.72 and an f1-score of 0.70 with word2vec embeddings. The CNN model has performed best for all the embeddings viz. word2vec, BERT, and fastText. The results of all the models in the experiment with the proposed architecture are shown in Table 3. Krishnamoorthy (Ma, 2019) proposed a hierarchical sentiment classifier (HSC) based on association rule mining which was able to achieve an accuracy of 0.83 and an F1-score of 0.81 on all agreed dataset of Financial Phrasebank.

Table 3
Comparison of different models with the proposed model.

Model	All agree on data		Merged Data	
	Accuracy	F1-score	Accuracy	F1-score
CNN (Word2Vec) (Mikolov et al., 2013)	0.79	0.76	0.72	0.70
BiLSTM (Word2Vec) (Mikolov et al., 2013)	0.68	0.65	0.64	0.64
CNN+BiLSTM (Word2Vec) (Mikolov et al., 2013)	0.71	0.69	0.67	0.66
CNN (BERT) (Devlin et al., 2018)	0.72	0.70	0.65	0.66
BiLSTM (BERT) (Devlin et al., 2018)	0.63	0.64	0.63	0.62
CNN+BiLSTM (BERT) (Devlin et al., 2018)	0.64	0.64	0.62	0.64
CNN (Fasttext) (Grave et al., 2018)	0.74	0.73	0.68	0.68
BiLSTM (Fasttext) (Grave et al., 2018)	0.69	0.70	0.66	0.66
CNN+BiLSTM (Fasttext) (Grave et al., 2018)	0.72	0.72	0.67	0.67
CNN (w2v+BERT)	0.84	0.84	0.82	0.82
BiLSTM (w2v+BERT)	0.83	0.83	0.78	0.77
CNN+BiLSTM (w2v+BERT)	0.82	0.84	0.82	0.81
Linearized Phrase Structure (LPS) (Malo et al., 2014)	0.79	0.80	0.71	0.71
Hierarchical sentiment classifier (HSC) (Ma, 2019)	0.83	–	0.71	–
Hybrid LPS (Štihec, Žnidaršič, & Pollak, 2018)	0.82	–	–	–
SVM (Unigram) (Ojo, Gelbukh, Calvo, Adebanji, & Sidorov, 2020)	–	–	0.77	0.76
Proposed Model	**0.86**	**0.86**	**0.83**	**0.83**

Similarly, Malo et al. (Malo et al., 2014) developed a linearized phrase structure model (LPS) specifically for the detection of contextual semantic orientations in the texts in the financial and economic domain. They implemented the model by adding a lexicon-based feature to the learning algorithm. The results from these experiments in the literature are also reported in Table 3.

4.2.2. Results and discussion

With the combination of BERT, word2vec, and POS on all agree data with data augmentation, the proposed CNN model performed better than other models with accuracy and f1-score of 0.86. Similarly, with the same CNN architecture, the best performance with an accuracy and f1-score of 0.83 was attained for merged data. Based on our experiments, the overall performances of the models have increased by a concatenation of the static and contextual embeddings. Moreover, the attention mechanism further boosted the performance of these models, implying that more weightage was given to those features which carried more importance in the sentence. Another thing to notice here is that CNN performed better in most the cases. The key thing to note here is that the CNN architecture contains max-pool filters after each convolution operation. This potentially extracts just the relevant features with reducing dimensionality simultaneously and eventually gives better performance. The CNN architecture showed increased performance when word2vec and BERT were merged which was further improved with the addition of POS tags. The proposed model achieved better performance scores than the other models discussed because it can tackle noise, semantics within the financial text, polysemy, and the context of finance domains. Further, the proposed model was able to outperform the past literature that used the same dataset of Financial phrase bank. The experiments were also carried out with FinBERT (Araci, 2019) and an accuracy of 0.63 was achieved. Table 3 also draws a comparison of the proposed methodology with the existing literature. The concatenation of static and contextual embeddings was useful as the static embeddings captured the frequency-related features of the word and contextual embeddings captured the context-related features of the words. These two features when concatenated gave a robust representation that included both context awareness and frequency-related themes. Similarly, POS features helped the representation to be better by the addition of part-of-speech-based features which are important to get the context of the words from the linguistic perspective. The hypothesis for this was that the embeddings would ideally benefit from this hybrid represen leading to a better representation helping and to create decision space for models. Similarly, attention mechanisms helped the model to maintain a focus on the representations that were important for models to classify the text. The max pooling layers helped to get a better test result by facilitating the model with an abstracted form of representation. Data augmentation on the other hand helped to deal with the problems of data imbalance. Thus, the process involving hybrid embeddings, attention mechanisms, and data augmentation helped the models to get a better decision space for the classification of the texts.

4.2.3. Ablation analysis

It can be seen from the results that combining the embedding layers improve the overall performance of the models. From Fig. 3, it is evident that when POS features are not considered, the performance drops slightly. From Table 4, it can be seen that the hybrid features with our model give an increase of over 15 percent accuracy. Furthermore, the experimental analysis also shows that a single embedding layer is not sufficient in giving good scores for the models. Hence, the proposed architecture propounds the idea that by including diversity in deep learning models by various feature extraction techniques, the models can capture the sense of the text and give a better performance for sentiment analysis. Moreover, from the experiments, it can also be observed that the models had a poorer performance with the original data, i.e., when there was a huge difference in the number of classes. With data augmentation, the scores have significantly increased. The word cloud of the most common words in positive, negative, neutral, and all the financial texts are shown in Fig. 4.

4.3. Explainability of the proposed models and embeddings

In our experimentation, a detailed study is done on how models perform with standard word representations and with hybrid word embeddings. The ablation study above shows that there is a significant improvement in the performances when hybrid word embeddings are used. The objective of this research is not just in the quantitative comparison of the performance measures but also in what is going on with the models. Explanations can help users to have much control over how the models are learning and hence users can build trust upon the model if it is learning in the right way. Also, with models outperforming humans in a lot of complicated tasks, the explainability of the models has become more important to make them more intelligent by understanding how a model comes to an outcome. To make things more explainable about what happens inside the black box models, the explainability and interpretability of the models with both hybrid and standard word representations are discussed. To interpret the models, Local Interpretable Model-agnostic Explanations

Table 4
Ablation study of different embeddings.

Model	All Agree		Merged Data	
	Accuracy	F1-Score	Accuracy	F1-Score
CNN+BiLSTM (Word2Vec)	0.71	0.69	0.67	0.66
CNN+BiLSTM (w2v+BERT)	0.82	0.84	0.82	0.81
CNN+BiLSTM (w2v+BERT+POS)	0.84	0.85	0.82	0.83
Proposed Model	**0.86**	**0.86**	**0.83**	**0.83**
Proposed Model without Augmentation	0.71	0.70	0.70	0.69

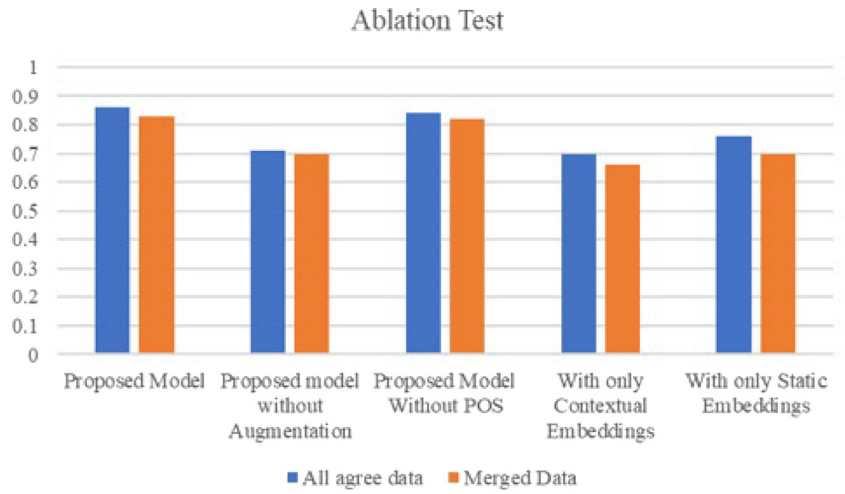

Fig. 3. Ablation test of the proposed model.

Fig. 4. Word cloud of (a) Positive, (b) Negative, (c) Neutral, and (d) All financial news.

(LIME) (Mardaoui & Garreau, 2021; Ribeiro, Singh, & Guestrin, 2016) have been used.

Fig. 5 shows the model's interpretation with and without hybrid embeddings. Fig. 5(a) shows the explanation of our proposed model with hybrid embeddings (POS + BERT + word2vec) whereas Fig. 5(b) shows the explanation of CNN + BiLSTM model with only BERT representations. When tested with an example, the model with hybrid embeddings shows the highest explanation for the words like 'drop', 'difficult', 'uncertainty', etc., and gives output as negative to the news annotated as negative. There is a prediction probability of 0.90 for the negative class. On the other hand, the CNN + BiLSTM model with BERT alone has given much weightage to words like 'to', 'said', 'profit', 'customers', 'drop', etc. and ultimately gave a label of positive to the news which is annotated as negative. It is noteworthy to see that the model is weak in classification with lower confidence scores. This shows that with hybrid word representations, the representations can decipher the in-depth information from the financial news and with single representation, the model is most likely to generalize the meaning of the word and not get into addressing the polysemy. Through analytical overview, the model using words like 'drop', 'difficulty', 'uncertainty', etc. to predict financial sentiment wins the trust of the users as the words 'drop', 'difficulty', 'uncertainty' is more associated with negative financial sentiments.

5. Conclusion

In this paper, a novel explainable hybrid word representation has been introduced that handles the hidden attributes and polysemic ambiguity of words. We used LIME to explain our predictions visually to indicate which words of the text contributed to the prediction. The proposed architecture can learn the semantics, sentiment, and syntactic differences of words within a financial statement and is also able to overcome the polysemy problem. By learning representations from three different feature extractors and attention to the neural network model, the model can outperform several baselines based on traditional word embeddings as well as contextual word embeddings when used separately. The experiments show that the fusion of different embedding methods significantly boosts the performances of the neural network models. Furthermore, it can also be inferred that the imbalanced class in the dataset hinders the overall performance of various models, and hence this should be handled using augmentation techniques. The explanation given by hybrid and single-word representations shows that the hybrid embeddings with rich representations can be highly efficient. In the future, we plan to incorporate character-level embeddings and a combination of them with the word embeddings. This helps to capture more features within a text to improve accuracy.

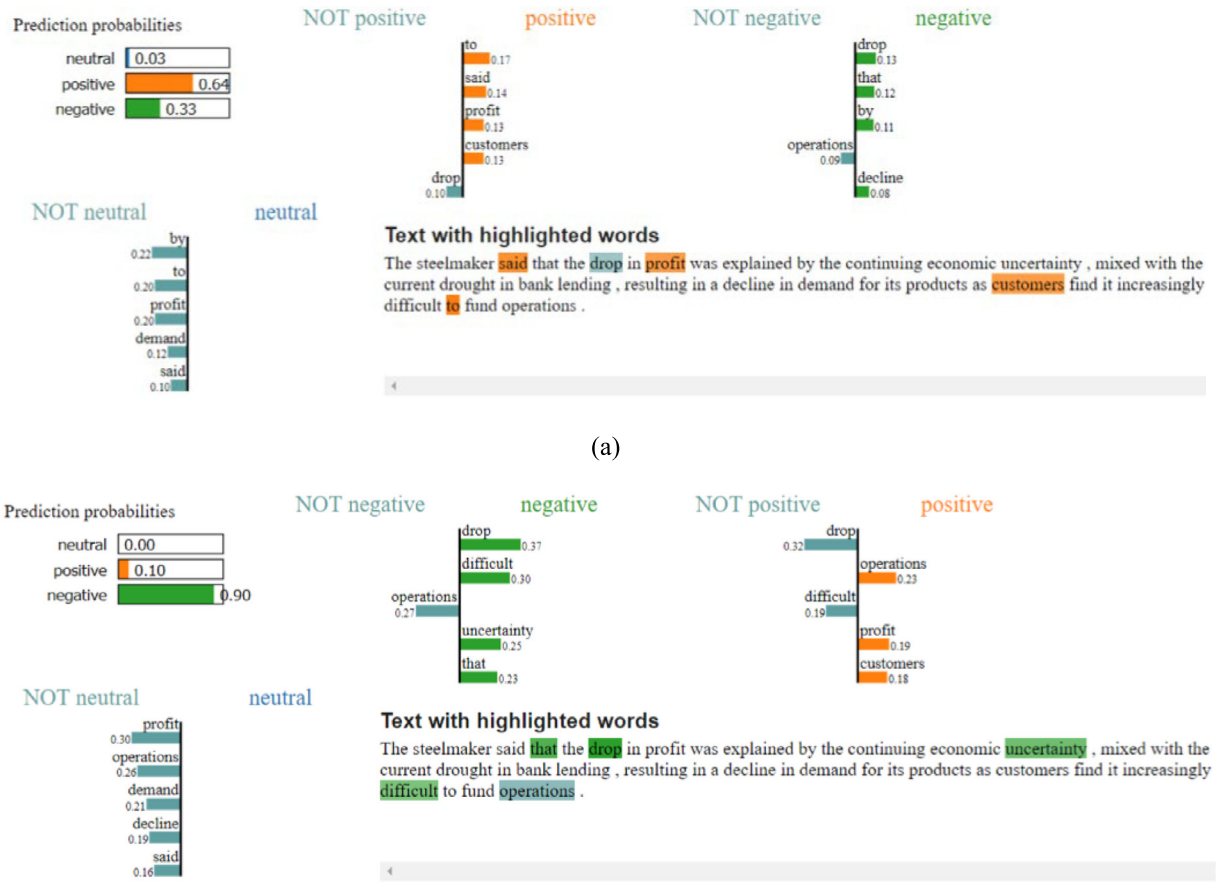

Fig. 5. Explainability of (a) model with BERT representations (b) model with hybrid word representations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Adhikari, S., Thapa, S., Singh, P., Huo, H., Bharathy, G., & Prasad, M. (2021). A comparative study of machine learning and NLP techniques for uses of stop words by patients in diagnosis of Alzheimer's disease. In *2021 international joint conference on neural networks*.

Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. arXiv [Cs.CL].

Arnold, T. M. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-ks. *C, F.A. Dig.*, *41*(2), 57–59.

Bai, H., Xing, F. Z., Cambria, E., & Huang, W.-B. (2019). Business taxonomy construction using concept-level hierarchical clustering. arXiv [Cs.CL].

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural probabilistic language models. In *Innovations in machine learning* (pp. 137–186). Berlin/Heidelberg: Springer-Verlag.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Choi, H., Cho, K., & Bengio, Y. (2018). Fine-grained attention mechanism for neural machine translation. *Neurocomputing*, *284*, 171–176.

Chowdhury, K. R., Sil, A., & Shukla, S. R. (2021). Explaining a black-box sentiment analysis model with Local Interpretable Model Diagnostics Explanation (LIME). In *International conference on advances in computing and data sciences* (pp. 90–101). Cham: Springer.

da Silva, N. F. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, *66*, 170–179.

Day, M.-Y., & Lee, C.-C. (2016). Deep learning for financial sentiment analysis on finance news providers. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv [Cs.CL].

Fersini, E., Messina, E., & Pozzi, F. A. (2014). Sentiment analysis: Bayesian ensemble learning. *Decision Support Systems*, *68*, 26–38.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. arXiv [Cs.CL].

Guo, L., Cheng, S., Liu, J., Wang, Y., Cai, Y., & Hong, X. (2022). Does social perception data express the spatio-temporal pattern of perceived urban noise? A case study based on 3, 137 noise complaints in Fuzhou, China. *Applied Acoustics*, *201*, Article 109129. http://dx.doi.org/10.1016/j.apacoust.2022.109129.

Gupta, A., Dengre, V., Kheruwala, H. A., & Shah, M. (2020). Comprehensive review of text-mining applications in finance. *Financial Innovation*, *6*(1).

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

Huang, C., Han, Z., Li, M., Wang, X., & W, Zhao. (2021). Sentiment evolution with interaction levels in blended learning environments: Using learning analytics and epistemic network analysis. *Australasian Journal of Educational Technology*, *37*(2), 81–95. http://dx.doi.org/10.14742/ajet.6749.

Huang, C., Jiang, F., Huang, Q., Wang, X., Han, Z., & Huang, W. (2022). Dual-graph attention convolution network for 3-D point cloud classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13. http://dx.doi.org/10.1109/TNNLS.2022.3162301.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv [cs.LG].

Kadlec, R., Schmid, M., Bajgar, O., & Kleindienst, J. (2016). Text understanding with the attention sum reader network. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*.

Kafle, K., Yousefhussien, M., & Kanan, C. (2017). Data augmentation for visual question answering. In *Proceedings of the 10th international conference on natural language generation*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing*.

Krishnamoorthy, S. (2018). Sentiment analysis of financial news articles using performance indicators. *Knowledge and Information Systems*, *56*(2), 373–394.

Li, Q., & 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*.

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, *337*, 325–338.

Liu, H., Liu, M., Li, D., Zheng, W., Yin, L., & Wang, R. (2022). Recent advances in pulse-coupled neural networks with applications in image processing. *Electronics*, *11*(20), http://dx.doi.org/10.3390/electronics11203264.

Liu, Y., et al. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv [Cs.CL].

Lu, H., Zhu, Y., Yin, M., Yin, G., & L, Xie. (2022). Multimodal fusion convolutional neural network with cross-attention mechanism for internal defect detection of magnetic tile. *IEEE Access*, *10*, 60876–60886. http://dx.doi.org/10.1109/ACCESS.2022.3180725.

Ma, E. (2019). *Nlpaug: data augmentation for NLP* (2019 ed.).

Malo, P., Sinha, A., Korhonen, P., Wallenius, J., & Takala, P. (2014). Good debt or bad debt: Detecting semantic orientations in economic texts: Good debt or bad debt. *Journal of Information Science and Technology Association*, *65*(4), 782–796.

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational linguistics and intelligent text processing* (pp. 171–189). Berlin, Heidelberg: Springer Berlin Heidelberg.

Mardaoui, D., & Garreau, D. (2021). An analysis of lime for text data. In *International conference on artificial intelligence and statistics* (pp. 3493–3501). PMLR.

Meng, F., Xiao, X., & Wang, J. (2022). Rating the crisis of online public opinion using a multi-level index system. *The International Arab Journal of Information Technology*, *19*(4), 597–608.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv [Cs.CL].

Naseem, U., Khan, S. K., Razzak, I., & Hameed, I. A. (2019). Hybrid words representation for airlines sentiment analysis. In *AI 2019: advances in artificial intelligence* (pp. 381–392). Cham: Springer International Publishing.

Naseem, U., & Musial, K. (2019). DICE: Deep intelligent contextual embedding for twitter sentiment analysis. In *2019 international conference on document analysis and recognition*.

Naseem, U., Razzak, I., Eklund, P., & Musial, K. (2020). Towards improved deep contextual embedding for the identification of irony and sarcasm. In *2020 international joint conference on neural networks*.

Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based deep intelligent contextual embedding for Twitter sentiment analysis. *Future Generation Computer Systems*, *113*, 58–69.

Ojo, O. E., Gelbukh, A., Calvo, H., Adebanji, O. O., & Sidorov, G. (2020). Sentiment detection in economics texts. In *Advances in computational intelligence* (pp. 271–281). Cham: Springer International Publishing.

Parihar, A. S., Thapa, S., & Mishra, S. (2021). Hate speech detection using natural language processing: Applications and challenges. In *2021 5th international conference on trends in electronics and informatics*.

Renault, T. (2020). Sentiment analysis and machine learning in finance: A comparison of methods and models on one million messages. *Digit Finance*, *2*(1–2), 1–13.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). 'Why should I trust you?': Explaining the predictions of any classifier. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: demonstrations*.

Saif, H., He, Y., & Alani, H. (2012). Semantic sentiment analysis of Twitter. In *The semantic web* (pp. 508–524). Berlin, Heidelberg: Springer Berlin Heidelberg.

Salazar, J., Kirchhoff, K., & Huang, Z. (2019). Self-attention networks for connectionist temporal classification in speech recognition. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing*.

Sehrawat, S. (2019). Learning word embeddings from 10-K filings using PyTorch. *SSRN Electronics Journal*.

Sheng, H., Cong, R., Yang, D., Chen, R., Wang, S., & Cui, Z. (2022). UrbanLF: A comprehensive light field dataset for semantic segmentation of urban scenes. *IEEE Transactions on Circuits and Systems for Video Technology*..

Soares Koshiyama, A., Firoozye, N., & Treleaven, P. (2020). Algorithms in future capital markets. *SSRN Electronics Journal*.

Sohangir, S., Wang, D., Pomeranets, A., & Khoshgoftaar, T. M. (2018). Big data: Deep learning for financial sentiment analysis. *Journal of Big Data*, *5*(1).

Štihec, J., Žnidaršič, M., & Pollak, S. (2018). *Lecture notes in computer science*, *Simplified hybrid approach for detection of semantic orientations in economic texts* (pp. 692–698). Cham: Springer International Publishing.

Sun, F., & Chu, N. (2020). Text sentiment analysis based on CNN-BiLSTM-attention model. In *2020 international conference on robots & intelligent system* (pp. 749–752). http://dx.doi.org/10.1109/ICRIS52159.2020.00186.

Thapa, S., Adhikari, S., & Mishra, S. (2021). Review of text summarization in Indian regional languages. In *Lecture notes in networks and systems* (pp. 23–32). Singapore: Springer Singapore.

Thapa, S., Adhikari, S., Naseem, U., Singh, P., Bharathy, G., & Prasad, M. (2020). Detecting Alzheimer's disease by exploiting linguistic information from Nepali transcript. In *Communications in computer and information science* (pp. 176–184). Cham: Springer International Publishing.

Vaswani, A., et al. (2017). Attention is all you need. arXiv [Cs.CL].

Wang, Qiqing, & Li, Cunbin (2022). Incident detection and classification in renewable energy news using pre-trained language models on deep neural networks. (pp. 57–76).

Wiebe, J. M., Bruce, R. F., & O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the association for computational linguistics on computational linguistics*.

Xing, F. Z., Cambria, E., Malandri, L., & Vercellis, C. (2019). Discovering Bayesian market views for intelligent asset allocation. In *Machine learning and knowledge discovery in databases* (pp. 120–135). Cham: Springer International Publishing.

Xiong, Z., Weng, X., & Y, Wei. (2022). SandplayAR: Evaluation of psychometric game for people with generalized anxiety disorder. *The Arts in Psychotherapy*, *80*, Article 101934. http://dx.doi.org/10.1016/j.aip.2022.101934.

Xu, Y., Qiu, X., Zhou, L., & Huang, X. (2020). Improving BERT fine-tuning via self-ensemble and self-distillation. arXiv [Cs.CL].

Yang, D., Zhu, T., Wang, S., Wang, S., & Z, Xiong. (2022). LFRSNet: A robust light field semantic segmentation network combining contextual and geometric features. *Frontiers in Environmental Science*, *10*, 1443. http://dx.doi.org/10.3389/fenvs.2022.996513.

Yeh, H.-Y., Yeh, Y.-C., & Shen, D.-B. (2020). Word vector models approach to text regression of financial risk prediction. *Symmetry (Basel)*, *12*(1), 89.

Yu, A. W., et al. (2018). QANet: Combining local convolution with global self-attention for reading comprehension. arXiv [cs.CL].

Zhao, N., Gao, H., Wen, X., & Li, H. (2021). Combination of convolutional neural network and gated recurrent unit for aspect-based sentiment analysis. *IEEE Access*, *9*, 15561–15569.