Sentence-level Sentiment Analysis Using GCN on Contextualized Word Representations

Huyen Trang Phan[1[0000−0002−7466−9562]], Ngoc Thanh Nguyen[2[0000−0002−3247−2948]], Zygmunt Mazur[2[0000−0003−1253−7895]], and Dosam Hwang[1,[0000−0001−7851−7323]]

[1] Department of Computer Engineering, Yeungnam University, Gyeongsan, South Korea
`huyentrangtin@gmail.com, dshwang@yu.ac.kr`

[2] Department of Applied Informatics, Wroclaw University of Science and Technology, Wroclaw, Poland
`Ngoc-Thanh.Nguyen@pwr.edu.pl, zygmunt.mazur@pwr.edu.pl`

Abstract. Sentiments expressed in opinions on social networks have played an increasingly significant impact in solving various social problems. Improving the effectiveness of sentiment analysis methods on social networks is still of interest to several scientists. A notable and robust development direction is sentiment analysis methods based on graph convolutional networks (GCNs). This paper introduces a model called contextual-based GCN for sentence-level sentiment analysis by considering the following steps: (i) Sentences are converted into contextualized word representation vectors based on the combination of the bidirectional encoder representations from the transformer model and bidirectional long short-term memory. (ii) The contextualized word representations are used to construct a sentence graph as a feature of nodes. (iii) A GCN model with two convolutional layers was used to learn the structure-aware node representations on the sentence graph. (iv) The softmax classifier was used for the sentence-level sentiment analysis. Experimental results on benchmark datasets showed that, unlike other methods, the proposed method can extract more context information from the opinions to obtain a better representation of the graph structure and learn better structure-aware nodes represented on the graph. The proposed method has improved the performance in terms of accuracy of the conventional methods from 2.2 to 3.2 percentage points.

Keywords: Sentence-level sentiment analysis · Graph Convolutional Network · BERT-BiLSTM · Contextual-based GCN.

1 Introduction

Since 2004, social networks have grown exponentially, and they have yet to reach the peak of their popularity. The latest statistics on social networks show that 3.78 billion users worldwide have used social media in 2021, and this number will continue to grow over the next few years[3]. Social media platforms are now

[3] https://www.statista.com/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

a significant source of news and information for researchers, governments, and businesses. By extracting and analyzing public moods and views on this data source, businesses, governments, and researchers can gain insight into trade, policies, and proposals, and make better decisions.

The sentiment analysis (SA) is a process of collecting, processing, analyzing, inferring, and synthesizing subjective sentiments contained in texts. SA of opinions is considered an exciting trend for artificial intelligence on social media. In the past, SA was considered a powerful tool for extracting and identifying the polarities of the emotions expressed in texts regarding opinions or evaluations about certain entities [13]. Today, as the economy is gradually transforming into a digital economy, SA is a significant step used in several systems automatically, such as stance detection, recommendation systems, decision-making, and fake news detection on social media. Three main levels of SA exist: document level, sentence level, and aspect level. Document-level SA aims to determine the sentiment polarity of the entire text without dividing sentences, and sentence-level SA seeks to determine the sentiment polarity of separate sentences. In comparison, aspect-level SA aims to determine the sentiment polarity regarding the aspects of entities appearing in the text. Sentence-level SA is the focus of this study. It is used to identify the sentiment in a sentence as positive, negative, or neutral. For instance, consider the sentence *"The phone color is not nice, but its style is so modern."* The sentiment in this sentence is positive.

Various approaches have been proposed for sentence-level SA, such as deep-learning-based, machine-learning-based, lexicon-based, and, most recently, graph convolutional network (GCN)-based approaches. GCN [11] is a graphically structured lattice and is a category of graph neural network. Consider a graph $G = \{V, E, A\}$, where V is a set of nodes; E is a set of edges, and A represent an adjacency matrix. GCNs use convolutional layers to learn node representations well by aggregating knowledge from the neighbours node on graph G [26]. They have achieved promising results in various tasks, particularly in natural language processing [1, 14]. The use of GCNs for SA has recently attracted considerable attention and has begun to produce expected results. However, previous GCN-based methods for SA often use GCNs with the limitations as follows.

- Most GCN-based methods focus on aspect-level SA, such as [11, 28]; rare approaches [2] consider sentence-level SA.
- Previous GCN-based sentence-level SA methods directly used GCNs without considering contextualized word representation.

The above two justifications motivated us to propose a novel sentence-level SA using GCN over contextualized word representations. The bidirectional encoder representations from the transformer (BERT) model is an embedding model that uses attention models as transformers to establish relationships between words via an encoder at the input and a decoder at the output. Unlike other embedding models that take one word at a time as input, BERT can take the entire sentence as input at once based on transformers. Therefore, BERT can learn the real meaning hidden between words [7]. Unlike basic grammar-based methods, only statistical characteristics that ignore context information

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

are considered. Bidirectional long short-term memory (BiLSTM) can learn contextual information, which is suitable for the logic of human language [3]. Meanwhile, GCNs have considerable expressive power for learning graph representations [29]. The proposed method includes the following steps: First, words in sentences are converted into word vectors, called BERT embeddings, using the pretrained BERT model [7]. Second, contextualized word vectors are created based on BiLSTM over BERT embeddings [8]. Third, the obtained vectors are used as word node features to build the sentence graph. Subsequently, the sentence graph is fed into the GCN model with two convolutional layers to create a sentence representation. Finally, the sentiment of the sentences is classified using the softmax classifier on the sentence representation. Experiments on two benchmark datasets illustrate that our proposed model showed an improvement in performance without using GCN over BERT-BiLSTM methods.

The remainder of this paper is organized as follows. The second section reviews related works on deep-learning-based and graph-based SA methods. The third section describes the research problem of this study. The fourth section presents the proposed method. The fifth section presents the experimental results and evaluations. The final section discusses the conclusions and future work.

2 Related Works

The previous approaches considered sentences in terms of types [20, 21], such as conditional and comparison sentences, or linguistics, such as a string of words with a dot symbol at the end. In this study, we considered sentences in terms of linguistics. This refers to considering sentences as short documents for the analysis. Deep-learning-based approaches are correctly oriented in the SA area, and using deep learning algorithms over graph structures is an interesting approach. In this section, we discuss some studies related to SA using deep-learning-based and graph-based methods.

The convolutional neural network (CNN) model was first proposed by Collobert for a semantic role labeling task [5]. In another attempt, Collobert [4] used a CNN model by serving a syntactic parse. A significant development in the direction of CNN-based SA methods is the study by Kim [9] regarding a simple CNN model with one convolution layer to create feature maps and until now it is often used as a strong baseline for various SA methods. Several variants have been introduced based on the CNN-based model of Kim, such as a densely connected CNN [27] for text classification. In addition, Poria et al. [23] applied a CNN for extracting document features and then fed them into a multiple-kernel learning model for SA. In another study [22], the authors used an extended long short-term memory (LSTM) for context information extraction from the surrounding sentences.

Graph structures were first used for SA methods as a representation step in various studies. Minkov et al. [17]built a labeled directed graph to represent words as nodes and the syntactic relation between the words as edges. They

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

then proposed a path-constrained graph walk SA method on the built graph. This algorithm performs better and is more scalable than other methods. Similarly, Violos et al. [26] presented an SA approach by extracting feature vectors from a word graph. Meanwhile, Bijari et al. [2] proposed a GCN-CNN model for sentence-level SA by considering semantic and term relations when text representation includes stopping words. This method marks a new direction for sentence-level SA using deep learning methods over graph structures.

The aforementioned studies show the performance of the deep learning and GCN methods for sentence-level SA. We consider whether combining GCNs and deep learning methods, such as BERT and BiLSTM, improves the performance of sentence-level SA. We aim to verify this hypothesis in this study.

3 Research Problem

3.1 Problem definition

Consider a set of n opinions $O = \{o_1, o_2, ..., o_n\}$. For $o_i \in O$, let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of sentences in opinions O. For $s_i \in S$, let $s = \{w_1, w_2, \ldots, w_m\}$ be a set of words of one sentence s_i and let c_i be contextualized word embeddings of sentence s_i. The objective of this study is to construct a GCN over the BERT-BiLSTM model for sentence-level SA. This objective can be formalized by finding a mapping function $F : (c_i) \to \{negative, neutral, positive\}$ such that:

$$F(c_i) = \begin{cases} positive, & if\ sentiment\ expressed\ in\ o_i\ is\ positive, \\ neutral, & if\ sentiment\ expressed\ in\ o_i\ is\ neutral, \\ negative, & if\ sentiment\ expressed\ in\ o_i\ is\ negative \end{cases} \quad (1)$$

3.2 Research questions

The main objective of this study is to propose a GCN for contextualized word representations (contextual-based GCN) for sentence-level SA. Therefore, we attempt to answer the following research questions:

- How can sentences be converted into contextualized word representations by combining the BERT and BiLSTM models?
- How can we build a sentence graph based on contextualized word representations?
- How can we construct a contextual-based GCN model by using a GCN with two convolutional layers over the sentence graph?
- How can the contextual-based GCN model be used for sentence-level SA?

4 Proposed method

In this section, we describe the concept and flow of the contextual-based GCN model. The proposed method is illustrated in Figure 1. The contextual-based GCN model consists of the following main steps:

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

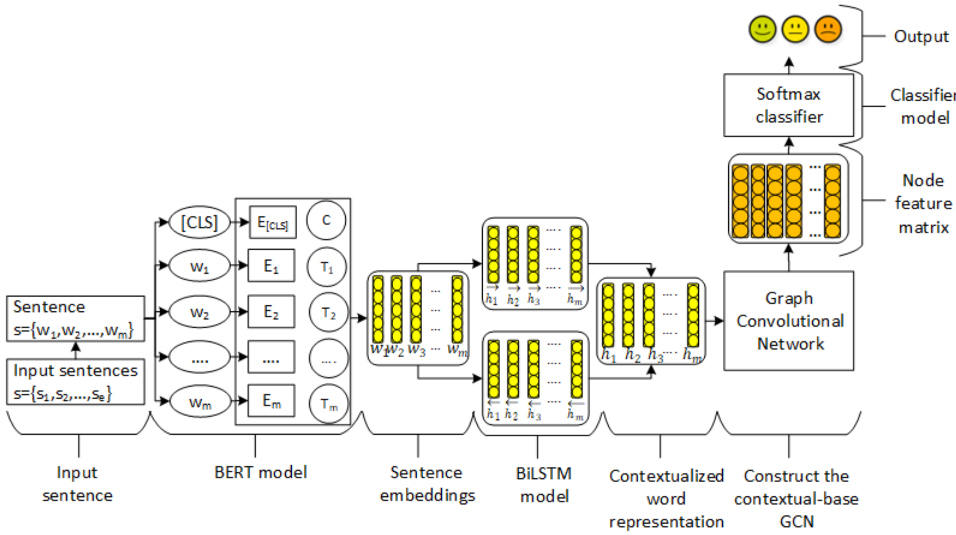

Fig. 1. Overall framework for proposed method

- Sentences are converted into contextualized word representation vectors based on the combination of the BERT and BiLSTM models.
- The contextualized word representations are used to construct a sentence graph as a feature of nodes.
- The GCN model with two convolutional layers is used to learn structure-aware node representations on the sentence graph.
- The softmax classifier is used for sentence-level SA.

The details of the approach are presented in the following sections.

4.1 Creating sentence vectors based on BERT

The BERT model automatically inserts a [CLS] symbol at the beginning of the sentence to indicate the beginning of each sentence. In this study, we intend to obtain the sentence vector; therefore, the BERT model uses the corresponding output vector to the symbol as the representation vector of the entire sentence as follows:

$$C = BERT(s) \in R^{m \times d_w} \quad (2)$$

where $BERT(s)$ is the vector corresponding to sentence s extracted from the pretrained BERT[4] [25], called BERT embeddings, and d_w is the dimension of the word vector.

[4] https://github.com/google-research/bert

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

4.2 Creating the contextualized word representations

In this study, we create the contextualized word representations using the BiLSTM model over the sentence vectors. Unlike basic grammar-based methods, only statistical features are considered, ignoring context information. BiLSTM can learn information about context, consistent with the logic of human language [3, 19]. The BiLSTM model was constructed based on the phases as follows:

Input layer: The BiLSTM model takes the sentence vector of the BERT model, where each word refers to a row of matrix C to which the weight matrix $W^a \in R^{d_a \times m}$ is added, as its input. This layer is formulated as follows:

$$a_i = \sigma(W^a.C_i + b^a) \in R^{d_a} \tag{3}$$

where σ is the sigmoid activation function. $i = [1, m]$, where m is the size of the sentence vector C_i, and b^a is the bias vector with dimension d_a.

BiLSTM layer: This layer learns the contextual information from the input layer via both directions for words. This layer consists of a forward LSTM and a backward LSTM to encode the sentence from left to right and vice versa, respectively. Therefore, from the word vector a_i, the BiLSTM layer creates a pair of hidden vectors $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ as follows:

$$\overrightarrow{h_i} = \overrightarrow{lstm}(a_i) \in R^{d_h}, i = [1, m] \tag{4}$$

$$\overleftarrow{h_i} = \overleftarrow{lstm}(a_i) \in R^{d_h}, i = [m, 1] \tag{5}$$

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{6}$$

where $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ are the hidden states of $\overrightarrow{lstm}$ and $\overleftarrow{lstm}$, respectively; h_i is the contextualized representation of the word w_i; for the i-th word in sentence s, $\overrightarrow{lstm}$ and $\overleftarrow{lstm}$ are the forward and backward LSTMs, respectively, and they are performed as follows:

For $\overrightarrow{lstm}(a_i)$:

$$\overrightarrow{G_i} = \begin{bmatrix} \overrightarrow{h}_{i-1} \\ a_i \end{bmatrix} \tag{7}$$

$$f_i = \sigma(W^f.\overrightarrow{G_i} + b^f) \tag{8}$$

$$in_i = \sigma(W^{in}.\overrightarrow{G_i} + b^{in}) \tag{9}$$

$$o_i = \sigma(W^o.\overrightarrow{G_i} + b^o) \tag{10}$$

$$c_i = f_i \odot c_{i-1} + in_i \odot tanh(W^c.\overrightarrow{G_i} + b^c) \tag{11}$$

$$\overrightarrow{h_i} = o_i \odot tanh(c_i) \tag{12}$$

where h_{i-1} is the previous hidden state of h_i and $h_0 = 0$.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

For $\overleftarrow{lstm}(a_i)$:

$$\overleftarrow{G_i} = \begin{bmatrix} \overleftarrow{h}_{i+1} \\ a_i \end{bmatrix} \tag{13}$$

$$f_i = \sigma(W^f.\overleftarrow{G_i} + b^f) \tag{14}$$

$$in_i = \sigma(W^{in}.\overleftarrow{G_i} + b^{in}) \tag{15}$$

$$o_i = \sigma(W^o.\overleftarrow{G_i} + b^o) \tag{16}$$

$$c_i = f_i \odot c_{i-1} + in_i \odot tanh(W^c.\overleftarrow{G_i} + b^c) \tag{17}$$

$$\overleftarrow{h_i} = o_i \odot tanh(c_i) \tag{18}$$

where σ is the sigmoid activation function. h_{i+1} is the next hidden state of h_i, and $h_{m+1} = 0$. c, o, in, f are the operations used in LSTM model. $\odot$ is element-wise multiplication. W^c, W^o, W^f, $W^{in} \in R^{d_h \times (d_h + d_a)}$, b^f, b^{in}, $b^o, b^c \in R^{d_h}$ are LSTM parameters, and d_h is the dimension of the hidden vectors.

Output layer: From the sentence vector matrix $C \in R^{m \times d_a}$, we obtain the contextualized word matrix $H = (h_1, h_2, ..., h_m) \in R^{m \times d_h}$, where d_h is the dimension of the contextualized vector.

4.3 Building the contextual-based GCN model

We build the contextual-based GCN model, which consists of the following two steps:

Building the sentence graph: A sentence graph is denoted by $G = (V, E, A)$, where V is a set of nodes, E is a set of edges, and A is an adjacency matrix. The nodes correspond to words in the sentence. The edges represent the dependencies of adjacent node pairs in the syntactic dependency tree[5]. The matrix $A \in R^{m \times m}$ represents the weights between nodes and are determined as follows:

$$A_{ij} = \begin{cases} 1, & if\ v_i, v_j \in V,\ and\ e_{ij} \in E, \\ 1, & if\ v_i = v_j, \\ 0, & otherwise \end{cases} \tag{19}$$

Additionally, graph G has a node feature matrix $Q = [H] \in R^{m \times d_h}$, where each row Q_i represents the contextualized vector of word node $v_i \in V$.

Creating the contextual-based GCN:Using the sentence graph, the contextual-based GCN is built by using two convolutional layers over the sentence graph as follows. Each node v_i on the sentence graph is represented by a vector $h_i \in R^{d_h}$. The entire graph yields a node feature matrix $Q \in R^{m \times d_h}$ and an adjacency matrix $A \in R^{m \times m}$. Subsequently, we feed matrices A and Q into a simple two-layer GCN proposed by Kipf et al. [11] as follows:

$$N^1 = \sigma(B \cdot N^0 \cdot D^1 + b^1) \tag{20}$$

[5] https://nlp.stanford.edu/software/stanford-dependencies.html

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

Hence,

$$N^2 = \sigma(B \cdot N^1 \cdot D^2 + b^2) \tag{21}$$

That means:

$$X = N^2 = \sigma(\sigma(B \cdot H^0 \cdot D^1 + b^1) \cdot D^2 + b^2)) \tag{22}$$

where $X \in R^{m \times d_h}$; $N^0 = Q$. σ is an activation function, such as $ReLU$. $D^1 \in R^{d_h \times m}$ and $D^2 \in R^{m \times d_h}$ are the weight matrices of two convolutional layers. b^1 and b^2 are the biases of two layers, respectively.

$$B = P^{-0.5} A P^{-0.5} \tag{23}$$

B is the symmetrically normalized matrix of A; P is the degree matrix of A, where:

$$P_{ii} = \sum_j A_{ij} \tag{24}$$

4.4 Sentence-level sentiment classifier

This step determines the sentiment polarity of a sentence based on the node representations. The sentiment classifier is defined as follows:

$$\hat{y} = \sigma(M \cdot X + b) \tag{25}$$

where σ is an activation function of Softmax. $M \in R^{l \times m}$ and $b \in R^l$ are a weight matrix and a bias of the σ, respectively, where l is the number of sentiment labels.

The GCN on the contextualized word representation model is trained by minimizing the cross-entropy error of the predicted and true label distributions using the following equation:

$$L = -\sum_i^l y_i log(\hat{y}_i) + \lambda \|\theta\|^2 \tag{26}$$

where y_i is the i-th real label distribution, and $\hat{y}_i$ is the i-th predicted label probability. λ is the coefficient of L_2 regulation. θ is the parameter set from the previous layers. The steps to train the GCN on contextualized word representations model are illustrated as Figure 2:

5 Experimental evaluation

5.1 Dataset and Experimental setup

In this study, to demonstrate the performance of our model and to ensure a fair comparison with other methods, we used benchmark datasets, such as IMDB

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

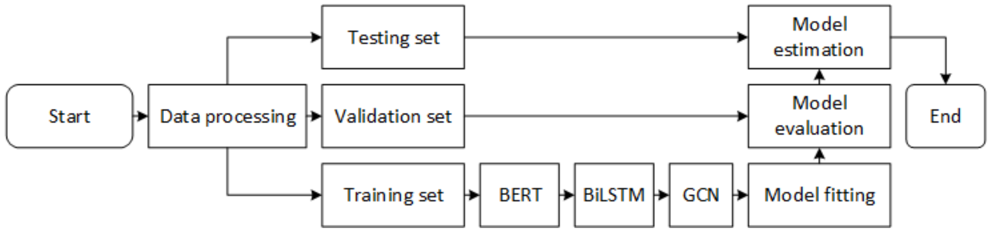

Fig. 2. The training process of the GCN over BERT-BiLSTM model

IMDB[6] and Financial PhraseBank[7] (FPB) are respectively from Kaggle[8]. Detailed information on the databases is shown in Table 1.

Table 1. Datasets used in experiments

Class	IMDB-Train	IMDB-Validation	IMDB-Test	FPB-Train	FPB-Validation	FPB-Test
Positive	4140	1035	5176	2558	639	320
Neutral	-	-	-	4862	1215	608
Negative	4134	1033	517	773	193	97
Total	8274	2068	5693	8193	2047	1025

The following parameters were set for the proposed model. For the BERT model, we used a pretrained BERT[9] and set the dimensions to 768. All the model weights were initialized with a uniform distribution. The dimensions of the hidden state vectors were set to 300. The Adam optimizer [10] was used with a learning rate of 0.001. The value of λ was 10^{-5}, and the batch size was 64. Moreover, the number of GCN layers was set to 2. The value of parameters was set up via the implementation process.

The results of the experimental process were obtained by averaging five-run results with random initialization, where the accuracy and loss were measured as the evaluation metrics [24]. We also compared the accuracy and loss with those of the baseline methods to confirm the improved performance of our proposed method.

6 https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews/version/1
7 https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-news/version/5?select=all-data.csv
8 https://www.kaggle.com/
9 https://github.com/google-research/bert

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

5.2 Baseline methods

To demonstrate the improved performance of our model compared with that of other models, we performed three different methods, namely, the proposed method and two baseline methods, on two datasets.

- The ensemble of LSTM and CNN [16] is an SA model based on LSTM to capture the temporal information and on CNN to extract the local structure from the data.
- CNN + GloVe and LSTM + GloVe [16] are variants of CNN and LSTM, respectively, using GloVe embeddings as the input layer.
- A modified GCN is a GCN-based SA with two convolutional layers.

5.3 Results and Discussion

The performances of the SA methods on the given datasets are shown in Tables 2 and 3.

Table 2. Performance of our proposal on Train and Validation (Val) datasets

IMDB						
Epoch	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
1	0.8850	0.2801	0.8504	0.3240	0.8366	0.3492
2	0.9402	0.1751	0.8704	0.3027	0.8512	0.3254
3	0.9631	0.1092	0.8759	0.3471	0.8579	0.3523
4	0.9850	0.0563	0.8686	0.3801	0.8579	0.4018
5	0.9931	0.0307	0.8723	0.4676	0.8546	0.4778
FPB						
Epoch	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
1	0.9240	0.2128	0.8704	0.3218	0.8839	0.2792
2	0.9757	0.0764	0.8869	0.3279	0.9580	0.1186
3	0.9872	0.0400	0.8631	0.4790	0.9733	0.0913
4	0.9887	0.0288	0.8686	0.5028	0.9713	0.1153
5	0.9939	0.0186	0.8723	0.4459	0.9726	0.0849

From Table 2, we can observe that, although the number of samples of the two datasets is the same, the performance of the proposed method on the IMDB dataset is slightly better than that on the FPB dataset. This is mainly because samples in the IMDB dataset are classified into two classes, whereas samples in the FPB are divided into three classes. This indicates that the samples in the IMDB dataset are denser than those in the FPB dataset. The proposed method can significantly improve this result by constructing a dataset that ensures a better balance between sentiment classes.

Table 3 presents a performance comparison of the models. The proposed method obtained better results than the baseline methods for the IMDB dataset.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

Table 3. Performance comparison of models on IMDB dataset (%)

Method	Accuracy
CNN + GloVe	89.3
LSTM + GloVe	89.0
Ensemble of LSTM and CNN	90.0
Modified GCN	90.0
Proposed method	92.2

The proposed method improved the accuracy of the LSTM + Glove model by 3.2 percentage points, that of the CNN + Glove model by 2.9 percentage points, and that of the LSTM + CNN model and the modified GCN model by 2.2 percentage points. Let us consider why the proposed method can enhance the accuracy of the baseline methods. In this study, the BERT model can capture the semantics of the text well, and the BiLSTM model can accurately extract the context of the sentence. Moreover, the combination of BERT and BiLSTM has considerable effectiveness in capturing contextualized representations. The results once again confirm that the use of GCN for contextual information representation significantly impacts the accuracy of sentence-level SA methods.

6 Conclusion and Future works

This paper proposed a method to improve the performance of sentence-level SA based on a GCN on a contextualized word representation model. Experimental results showed that the proposed method significantly improves the performance of sentence-level SA on two benchmark datasets. However, this method does not consider all contextual factors, semantic relations, and emotional knowledge simultaneously when building text representation graphs. In the future, we will focus on the following directions: (i) Building graphs simultaneously represent contextual factors, semantic relations [18], and sentimental knowledge to improve the performance of SA methods. (ii) Constructing a consensus-based [6], user profiles tuning-based [15], and symbolic knowledge representation-based [12] GCNs for ALSA are also an interesting approaches.

References

1. Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., Sima'an, K.: Graph convolutional encoders for syntax-aware neural machine translation. arXiv preprint arXiv:1704.04675 (2017)
2. Bijari, K., Zare, H., Kebriaei, E., Veisi, H.: Leveraging deep graph-based text representation for sentiment polarity applications. Expert Systems with Applications **144**, 113090 (2020)
3. Cai, R., Qin, B., Chen, Y., Zhang, L., Yang, R., Chen, S., Wang, W.: Sentiment analysis about investors and consumers in energy market based on bert-bilstm. IEEE Access **8**, 171408–171415 (2020)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

4. Collobert, R.: Deep learning for efficient discriminative parsing. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 224–232. JMLR Workshop and Conference Proceedings (2011)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of machine learning research **12**(ARTICLE), 2493–2537 (2011)
6. Daniłowicz, C., Nguyen, N.T.: Consensus-based methods for restoring consistency of replicated data. In: Intelligent Information Systems, pp. 325–335. Springer (2000)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks **18**(5-6), 602–610 (2005)
9. Kim, Y.: Convolutional neural networks for sentence classification. CoRR **abs/1408.5882** (2014), http://arxiv.org/abs/1408.5882
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Kolaczek, G., Pieczynska-Kuchtiak, A., Juszczyszyn, K., Grzech, A., Katarzyniak, R.P., Nguyen, N.T.: A mobile agent approach to intrusion detection in network systems. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. pp. 514–519. Springer (2005)
13. Liu, B.: Sentiment analysis and opinion mining. Synthesis lectures on human language technologies **5**(1), 1–167 (2012)
14. Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. arXiv preprint arXiv:1703.04826 (2017)
15. Mianowska, B., Nguyen, N.T.: Tuning user profiles based on analyzing dynamic preference in document retrieval systems. Multimedia tools and applications **65**(1), 93–118 (2013)
16. Minaee, S., Azimi, E., Abdolrashidi, A.: Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models. arXiv preprint arXiv:1904.04206 (2019)
17. Minkov, E., Cohen, W.: Learning graph walk based similarity measures for parsed text. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. pp. 907–916 (2008)
18. Nguyen, N.T.: Processing inconsistency of knowledge on semantic level. J. Univers. Comput. Sci. **11**(2), 285–302 (2005)
19. Phan, H.T., Nguyen, N.T., Hwang, D.: Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis. Information Sciences (2022)
20. Phan, H.T., Nguyen, N.T., Tran, V.C., Hwang, D.: A method for detecting and analyzing the sentiment of tweets containing conditional sentences. In: Asian Conference on Intelligent Information and Database Systems. pp. 177–188. Springer (2019)
21. Phan, H.T., Nguyen, N.T., Van Cuong, T., Hwang, D.: A method for detecting and analyzing the sentiment of tweets containing fuzzy sentiment phrases. In: 2019 IEEE international symposium on innovations in intelligent systems and applications (INISTA). pp. 1–6. IEEE (2019)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71

22. Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A., Morency, L.P.: Context-dependent sentiment analysis in user-generated videos. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers). pp. 873–883 (2017)
23. Poria, S., Chaturvedi, I., Cambria, E., Hussain, A.: Convolutional mkl based multimodal emotion recognition and sentiment analysis. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 439–448. IEEE (2016)
24. Schütze, H., Manning, C.D., Raghavan, P.: Introduction to information retrieval, vol. 39. Cambridge University Press Cambridge (2008)
25. Turc, I., Chang, M.W., Lee, K., Toutanova, K.: Well-read students learn better: On the importance of pre-training compact models. arXiv preprint arXiv:1908.08962v2 (2019)
26. Violos, J., Tserpes, K., Psomakelis, E., Psychas, K., Varvarigou, T.: Sentiment analysis using word-graphs. In: Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics. pp. 1–9 (2016)
27. Wang, S., Huang, M., Deng, Z., et al.: Densely connected cnn with multi-scale feature attention for text classification. In: IJCAI. pp. 4468–4474 (2018)
28. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 7370–7377 (2019)
29. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. Computational Social Networks **6**(1), 1–23 (2019)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_71