Information Sciences 589 (2022) 416–439

Contents lists available at ScienceDirect

Information Sciences

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

Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis

Huyen Trang Phan [a,1], Ngoc Thanh Nguyen [b,*,2], Dosam Hwang [a,*,3]

[a] Department of Computer Engineering, Yeungnam University, Gyeongsan, Republic of Korea
[b] Department of Applied Informatics, Wroclaw University of Science and Technology, Wroclaw, Poland

ARTICLE INFO

Article history:
Received 14 May 2021
Received in revised form 30 December 2021
Accepted 30 December 2021
Available online 06 January 2022

Keywords:
SSC graph
CANN-SSCG model
Aspect-level sentiment analysis
Syntax-based graph
Semantic-based graph
Context-based graph

ABSTRACT

Recently, aspect-level sentiment analysis methods using graph convolutional network (GCN)-based structures with fairly good performance have been introduced. However, previous GCN-based methods often experience one of the following limitations. First, GCNs usually use edges with binary weights. However, binary weights are not helpful in many tasks. Second, these GCNs only focus on extracting node features from some single words or phrases and ignore their context in the entire sentence or paragraph or only consider the information of independent phrases when determining the relation between two graph edges overlooking the semantic relation among these phrases. Finally, no studies simultaneously use the information on the context, the semantic relation, and the sentiment knowledge among words or phrases to build GCNs for aspect-level sentiment analysis. Therefore, to resolve these limitations, in this study, we propose a new method, the CANN-SSCG model, as follows. First, we built three separate heterogeneous graphs, namely, syntax-based, semantic-based, and context-based graphs. Second, we constructed a general heterogeneous graph (SSC graph) by combining the three constructed graphs. We then converted the nodes of the SSC graph into sentence vectors using a GCN with two layers (creating an SSC-GCN). Finally, we used a convolutional neural network algorithm with attention to position embeddings (CANN) on the output of the SSC-GCN model for aspect-level sentiment analysis. The experiments, which used three different datasets, including reviews and tweets, showed that the proposed method yields promising results based on the F_1*score*.

© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Sentiment analysis, a specific task in the field of natural language processing, is a process to determine the polarity of a user's emotions in opinions regarding entities or their aspects and measures how positively or negatively the user of an entity or entity's aspect is regarded [32,50]. The entities included in opinions can refer to people, organizations, events, loca-

* Corresponding authors.
E-mail addresses: Ngoc-Thanh.Nguyen@pwr.edu.pl (N.T. Nguyen), dshwang@yu.ac.kr (D. Hwang).
[1] ORCID ID: https://orcid.org/0000-0002-7466-9562
[2] ORCID ID: https://orcid.org/0000-0002-3247-2948
[3] ORCID ID: https://orcid.org/0000-0001-7851-7323

https://doi.org/10.1016/j.ins.2021.12.127
0020-0255/© 2022 The Author(s). Published by Elsevier Inc.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

tions, products, or topics. An entity is often a noun or noun phrase; and an entity is typically described in more detail by clauses or sentences regarding aspects pertaining to it. An aspect is a subpart, a specific detail, or an attribute of an entity; and an aspect is typically expressed by a noun or noun phrase that is complemented by a phrase, clause, or sentence expressing a user's emotion. Sentiment analysis (SA) in texts can be divided into levels such as the document level, sentence level, and aspect level. Document-level SA identifies the emotional polarity (e.g., negative, neutral, or positive) of the entities in an entire document. Sentence-level SA identifies the sentiment polarity of the entities in each sentence. Aspect-level SA is used to determine the human emotional orientation of the attributes or parts of entities appearing in a document [10,11]. The aspect-level SA of personal reviews of products, services, events, and policies is widely applied in various practical tasks, especially decision-making applications, recommendation systems, and fake news detection [50]. For example, Phan et al. [32] presented a novel method based on aspect-level SA to determine the degree of user satisfaction for a decision-making support system. Oppong et al. [28] built a system, namely, SentDesk, that is able to support a business by considering sentiments in users' opinions. Therefore, aspect-level SA has been widely investigated by many researchers. Its main challenge is determining the sentiment relationship between one word or phrase expressing an aspect and words describing the sentiment of a word or phrase indicating an aspect correctly. To date, this challenge has yet to be completely solved. As the improvement of the performance of aspect-level SA methods is a critical issue, many methods have been developed to increase the efficacy of the aspect-level SA of opinions using deep learning algorithms, statistical learning techniques, and, most recently, GCN structures. The advantages and disadvantages of these methods are listed in Table 1.

As shown in Table 1, the accuracy of the graph-based and deep learning methods is good, and their training time is medium. Therefore, different combination approaches have been proposed, and the newest methods use deep learning algorithms over graph structures [41,35]. More recently, the combination of deep learning algorithms over the GCN model has been studied. This is an exciting topic in machine learning and other areas. Many methods based on GCNs have been proposed and widely used for predicting personal relationships in social networks. These methods have improved the prediction of recommendation systems and the segmentation of large point clouds [19], object detection, fake news detection, and machine translation. GCNs [19] are graphically structured lattices and are one category of graph neural network models. Given a graph $G = \{V, E, A\}$ including a set of nodes V, a set of edges E, and the adjacency matrix A, the objective of GCNs is to use convolutional network layers to learn node representations well by integrating information from the neighbourhoods (node features and edge weights) on the given graph [22]. If there is one layer, a GCN can only represent nodes using neighbouring information. Otherwise, if there are multiple layers, a GCN can represent nodes using more neighbourhood information. GCNs have achieved promising performance in many tasks. However, previous GCN-based approaches often include one of the following limitations:

- GCNs usually use edges with binary weights (i.e., 0 and 1). However, binary weights are not helpful in many actual tasks. For example, assume we want to find groups containing edges with very low weights or high weights in the clustering task. In this case, if the weights of edges are binary values, the given problem cannot be solved by conventional GCNs [42].
- They only focus on extracting node features from some single words or phrases and ignore their context in the entire sentence or paragraph or only consider the information of independent phrases when determining the relation between two graph edges overlooking the semantic relation among these phrases. For example, take the sentence "*The king crab soup is not too fresh*". This opinion includes a fuzzy sentiment phrase of three words, namely, "*not too fresh*". However, some approaches classify the sentiment as positive due to only considering the impact of the word "*fresh*", ignoring the effect of both "*not*" and "*too*", while some other approaches classify the sentiment as negative because they ignore the impact of the word "*too*".
- Information on the context, semantic relations, and sentiment knowledge among words or phrases is essential and helpful when constructing GCNs for aspect-level SA. However, only a limited number of studies have explored incorporating some of these features via flexible GCNs for this task. For example, in [50], the authors modelled syntax and knowledge via a GCN. In [49], the authors used GCNs to create a single text graph based on word co-occurrence and document-word relations. This means that to our knowledge, there are no GCNs considering all of the context, the semantic relations, and the sentiment knowledge factors simultaneously.
- Last, very few approaches use convolutional neural network (CNN)-extended algorithms over GCNs with state-of-the-art performance, such as [2,47]. However, they either focus on document-level SA without considering aspect-level SA or they do not apply these proposals to text data. In addition, they do not consider adding an attention mechanism to the CNN algorithm.

Therefore, the construction of a general heterogeneous graph by integrating syntax-based, semantic-based, and context-based graphs and using a CANN on this graph can solve the above limitations of aspect-level SA methods. For these reasons, we developed a methodology to improve the performance of aspect-level SA via the convolutional attention neural network (CANN) algorithm on graph structures (referred to as CANN-SSCG). The proposed method consists of four steps. First, contextualized word representations are determined using bidirectional long short-term memory (BiLSTM) on GloVe embeddings. Second, a syntax-based graph, a semantic-based graph, and a context-based graph are constructed and integrated into a general graph, called the SSC graph. Third, nodes on the SSC graph are fed into two layers of the GCN model (called the SSC-GCN model) to convert them into a vector space by aggregating the node feature matrices obtained from the three

Table 1
Comparison advantages and disadvantages of sentiment analysis approaches [12].

Approach	Time training	Accuracy	Advantage	Disadvantage
			Machine learning	
Supervised learning	Slow	High	Can be applied to both binary SA and multiple SA Helpful to determine subjective opinions Less affected by noise data	Highly affected by labeled data Take a lot of time and cost
Semi-supervised learning: Self-training Co-training Graph-based Multi-view learning Generative models	Medium	High	Can be high performance with limited labeled data Can solve the cross-lingual problem	Highly affected by noise data Highly dependent on extracted features Poor performance with a limited vocabulary of words Inflexibility
Unsupervised learning	Fast	Medium	Not take much time and cost for labeling data Obtain high efficiency and can handle various tasks	Unstable performance Poor performance with the multiple SA Obtain low efficacy with noise data
Deep learning: Autoencoder neural network Convolutional neural network Long short term memory Recursive neural network	Slow	High	Can be applied to both binary SA and multiple SA Useful to determine subjective opinions Less affected by noise data Can extract features automatically	Most affected by labeled data Take a lot of time and cost
			Lexicon-based	
Dictionary-based	Very fast	Relatively low	Not need labeled data Simple implementation	Have no full lexicons of words Not fit with the implicit sentiment analysis Low performance with aspect-level SA
Corpus-based	Very fast	Relatively low	Fit to content-oriented data Can obtain high performance with multi-domains opinions	Highly dependent on the lexicon domains Poor performance with large documents Low efficiency without combining with other methods
			Hybrid approach	
Hybrid method	Medium	High	Can be flexible to use Can leverage the advantages of combined methods	Difficult to find suitable combining methods

built graphs. Finally, the positions of aspects and their sentiments are determined using the CANN model by attending the position embeddings to the SSC-GCN structure.

The new points of the CANN-SSCG model in comparison to other structures are as follows:

- This method is a combination of a graph structure and deep learning method, so it can take advantage of both of them and create a model with more than two layers.
- This method improves the conventional graph structure with edges with arbitrary weights.
- This method also improves the conventional deep learning method by adding the attention mechanism to the position embeddings of the words in a given sentence.

Compared to other tools, the CANN-SSCG can more precisely extract the essential features related to aspects and their sentiment when representing sentences appearing in a given opinion. Therefore, this method can determine the aspects and classify the sentiment of the aspects more precisely. The main contributions of this proposal can be summarized as follows:

- We show that the SSC-GCN is powerful for sentence representation. A CANN adding an attention mechanism for the positions of words indicating aspects is effective in aspect-level sentiment classification. The combination of an SSC-GCN and a CANN is significant for improving the performance of aspect-level SA.
- We construct three novel efficient graph-based architectures: a syntax-based graph, a semantic-based graph, and a context-based graph. The node feature of a syntax-based graph is a word; that of a syntax-based graph is contextualized word representations; and that of a semantic-based graph is the combined representations among contextualized word representations, word position embeddings, and common sense knowledge embeddings. The node features of the context-based graph are the context embeddings of words.

- We integrate a syntax-based graph, a semantic-based graph, and a context-based graph into a new heterogeneous graph called the SSC graph to learn the most important information from the previous three graphs for sentence representation. Here, the node features of the SSC graph are the concatenation vector of the contextualized word vector, common sense knowledge vector, sentiment score vector, and context-specific vector. In particular, the edges of the SSC graph have weights that are not binary values.
- We use the CANN algorithm on the SSC graph to build a new aspect-level sentiment classification model, called the CANN-SSCG model, to determine the positions of aspects and their sentiment for aspect-level sentiment classification. The CANN-SSCG model with more layers than the conventional GCNs and integrating the position information of words indicating aspects of the attention mechanism achieves better predictions of the target aspects and their sentiments.

The remainder of this paper is organized as follows. In Section 2, we present an overview of the related aspect-level SA. The research problem of the proposed method is described in Section 3. In Section 4, we present a mathematical model to solve the research problem. A brief introduction to the datasets and experimental results of the proposed approach versus some well-known methods is provided in Section 5. The conclusions and future work are presented in Section 6.

2. Related works

2.1. Aspect-level sentiment analysis

Aspect-level SA aims to classify the sentiment polarity of a target in a given context. He et al. [11] introduced a novel model that combines the syntax information obtained from a tree, depending on the attention mechanism. Most previous studies used long short-term memory (LSTM) and attention mechanisms to classify the sentiment of aspects, which usually have high complexity and take considerable time to train the models [36]. Some previous methods used the CNN method and gating mechanisms, which have lower complexity and take less time to train but are more efficient than LSTM because the computations are parallelized throughout training the model. However, such CNN-based approaches do not consider the separate modelling of target aspects through representations of context-specific information. To overcome these drawbacks, Kumar et al. [20] provided a novel interactive gated convolutional network using a bidirectional gating mechanism to learn the mutual relations between a target aspect and a corresponding opinion context. However, the method ignores the impact of comparative adjectives, words that express needs or wants, and noncompositional expressions. Wu et al. [45] presented an interactive representation method between aspects and contexts that relies only on an attention mechanism. However, this model disregards the order of aspects and sequence of contexts. Nguyen et al. [26] proposed an interactive network based on lexicon awareness and aspect attention. In this model, the authors combined interactive attention and intraattention mechanisms with the lexicon containing sentiment information to represent aspect-specific information at both the phrase and aggregation levels. Shuang et al. [40] proposed a feature extraction network based on reducing noise and unnecessary features by designing a double-gate mechanism to determine the interactions between aspects and their corresponding contexts. As the previous research shows, the original methods mainly applied traditional machine learning methods to classify sentiments at the aspect level [10,11,32]. However, machine learning methods, including deep learning, depend heavily on extended handcrafted features [10]. Neural networks have recently achieved outstanding performance in aspect-level SA because they can automatically learn distributed representations of contextual words [2,33]. In addition, some researchers used recursive neural networks to construct the binary tree of sentences using a dependency tree [24,25]. In addition, some aspect-level SA methods integrated the location information of target aspects [32,50]. Attention mechanisms have been adopted to enforce models that focus on the important parts related to the given aspects. By paying attention to the number of contextual words and the dependency tree structure, some authors [5,27] have taken advantage of the positions of words indicating aspects and the surrounding contextual words. Furthermore, Phan et al. [32] incorporated the general knowledge of concepts related to emotions, such as fuzzy sentiment phrases and clear sentiment phrases, into a BiLSTM model to classify emotions at the aspect level.

2.2. Convolutional attention neural network

The convolutional neural network algorithm was proposed by LeCun et al. [21] and successfully used in natural language processing [17,47]. The CANN [36,46] is an improvement of the CNN model in that it applies an attention mechanism, which has been shown to improve performance in machine translation [39], entailment reasoning [37], sentence summarization [38], and SA [46]. In [48], the authors introduced an improved BiLSTM method by considering two attentions. In [46], the authors presented a new method for aspect-level SA by integrating attention-based input into the CNN model to determine contextual words. Furthermore, in [36], the authors introduced an approach for mining aspect-based opinions by introducing an attention mechanism into the pooling layer of a CNN model. Wang et al. [43,44] provided two methods for improving the efficacy of text classification. The first method [44] presented an LSTM model with attention to the significant terms in a sentence, and the second method [43] concatenated a densely connected CNN with multiscale features for classifying texts. Another case of CNN improvement is to use a CNN algorithm on graphs, such as the graph convolutional neural network provided by Yan et al. [47] to classify building patterns with high performance; and using a CNN on graphs to perform fast localized spectral filtering was introduced by Defferrard et al. [7]. Therefore, a CNN over a GCN has been used effectively for image datasets. This suggests that this combination would also yield state-of-the-art results for text datasets.

2.3. Graph convolutional networks

Various methods have been introduced to embed a network to understand the nodes (with low-dimensional vectors) in a graph network. Recently, graph neural networks (GNNs) have quickly gained notice owing to their great expressive power [19]. A GCN is a variation of a GNN and was first introduced by Kipf and Welling [19], and they obtained good performance on the text classification task on some benchmark datasets. GCNs have also been explored in some NLP tasks, such as semantic role labelling [25], relational classification [50], text classification [2], and machine translation [24]. In [29], the authors presented a deep learning model using CNN graphs to convert text into word graphs. Then, in [27], the authors applied convolution operations to the graphs to transform the word graphs. In [49], the authors commented on the text and the nodal words to build a heterogeneity graph (called TextGCN) and used this GCN to represent words and documents; and in [25], the authors introduced a GCN syntax that is a direct graph with labelled edges for aspect-level SA with quite good accuracy. Bastings et al. [1] also used the syntactic GCN for syntax-aware tasks. Furthermore, in [24], the authors used the syntactic GCN to combine the predicate information and the argument structure of a target sentence. Zhou et al. [50] introduced an improved GCN method for aspect-level sentiment classification. This method focused on constructing two separate graphs, namely, the syntax graph and the knowledge graph. Then, two built graphs were integrated into one GCN to classify the sentiment of aspects. Bijari et al. [2] presented a new approach for aspect-level SA by leveraging a deep-based text representation. This method used combined sentence graphs to collect the continuous and latent features of the documents. Then, these features were fed into a deep neural network to determine the sentiment of aspects. Their method can be used on various datasets without concerning pretrained word embeddings. The experimental results prove that the previous method that we analyse above achieves good performance on benchmark datasets.

The analysis of the previous literature shows that many researchers have aimed to improve the efficacy of aspect-level SA methods by taking advantage of deep learning algorithms and GCNs. However, most previous approaches only used these methods individually and only rarely created models that combined them into one general process to identify aspect-level sentiment. In addition, the syntactic representation tree, knowledge representation tree, and context representation tree are graph data structures; therefore, it is natural to apply a GCN to encode them. Therefore, this study proposes a new GCN structure to model syntax, semantic, and contextual information. The CANN model is then used over the constructed GCN for aspect-level SA.

3. Research problem

3.1. Background definition

Definition 1. A sentiment relation [32] between word tokens w_k and w_h (w_h expresses the sentiment and w_k indicates the aspect) is defined by the following function, Θ:

$$\Theta(w_k, w_h) = \begin{cases} 1, & \text{if } w_k \text{ related to } w_h \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Definition 2. An aspect [32] a of a specified entity in opinion t is a token w_k that satisfies two conditions simultaneously: w_k must be a noun or noun phrase and must be related to at least one sentiment word existing in this opinion. The aspect a is expressed as the following equation:

$$a = \{w_k | tag(w_k) = \text{`}NOUN\text{'}, \exists w_h \in t : \Theta(w_k, w_h) = 1\} \tag{2}$$

Definition 3. The sentiment of aspect a_i in sentence s, denoted by se_i, is defined as follows:

$$se_i = \begin{cases} Positive, & \text{if } label(a_i) = \text{`}aspect\text{'} \wedge \exists w \in s : label(w) = \text{`}positive\text{'} \wedge \Theta(w, a_i) = 1 \\ Neutral, & \text{if } label(a_i) = \text{`}aspect\text{'} \wedge \exists w \in s : label(w) = \text{`}neutral\text{'} \wedge \Theta(w, a_i) = 1 \\ Negative, & \text{if } label(a_i) = \text{`}aspect\text{'} \wedge \exists w \in s : label(w) = \text{`}negative\text{'} \wedge \Theta(w, a_i) = 1 \end{cases} \tag{3}$$

3.2. Problem definition

Take a finite set of opinions, T, where $T = \{t_1, t_2, \ldots, t_n\}$, and n is the number of gathered opinions. Assume that each opinion t contains some sentences $S = \{s_1, s_2, \ldots, s_e\}$, where each sentence s includes different sentiments regarding several aspects. Let $A = \{a_1, a_2, \ldots, a_h\}$ be a set of aspects appearing in opinions T; and let $Se = \{se_1, se_2, \ldots, se_e\}$ represent a set of sentiments related to aspects, where $se_i \in \{positive, neutral, negative\}$. This study aims to solve the following task. Performing grained SA on the aspects in opinions means identifying the set of sentiments of aspects $Se = \{se_1, se_2, \ldots, se_h\}$. The formal task is described as follows:

$$se_i = Frame(t(s)) \tag{4}$$

where se_i refers to the sentiment of aspect a_i in sentence s of opinion $t; Frame(.)$ represents the frame of CANN-SSCG model that is proposed; $t(.)$ represents an opinion with multiple sentences (number of sentences $\geqslant 1$), and s represents a sentence in the opinion with at least one target aspect and its sentiment (number of target aspect's sentiment $\geqslant 1$).

3.3. Research question

In this paper, to solve the research problem presented in Section 3.2, we seek to find the answer to the main question: How can we use the CANN on graph structures to improve the performance of aspect-level SA? This question can be divided into the following four subquestions:

(i) How can we construct a syntax-based graph, a semantic-based graph, and a context-based graph?
(ii) How can the three built graphs be integrated into a new graph (called the SSC graph)?
(iii) How can the CANN algorithm be applied to the SSC graph (i.e., how can the CANN-SSCG model be built)?
(iv) How can the CANN-SSCG model be used to classify the sentiment of aspects appearing in opinions?

4. Mathematical model of the proposed method

The goal of this study is to prove that the CANN-SSCG model can improve the performance of conventional GCNs in aspect-level SA. This section introduces a methodology to answer the research questions shown in Section 3. Our proposal includes four stages as follows.

(i) Contextualized word representation: This step aims to determine the feature-word nodes of graphs by using BiLSTM on the GloVe embeddings.
(ii) Graph construction: This is the first step in building syntactic-based, semantic-based, and context-based graphs. These graphs are then integrated into an SSC graph.
(iii) Node representation: This step converts feature word nodes on the SSC graph into sentence vectors by feeding the SSC graph into the two-layer GCN to create the SSC-GCN model.
(iv) Aspect-level sentiment classification: This step builds the CANN-SSCG model with the CANN algorithm by adding position embeddings on the SSC-GCN structure for aspect-level SA.

For the convenience of following the proposed method step by step, we list the notations used in this paper in Table 2. The workflow of the method is shown in Fig. 1.

4.1. Contextualized word representation

In this study, we use the BiLSTM model [13] to represent the contextualized words. The contextualized word representations are the word vectors that are more sensitive to the context in which words appear. In this paper, the contextualized word representations are based on a BiLSTM model. The BiLSTM model can fuse the contextual dependence relationships. In our proposal, BiLSTM is used to integrate the contextual information by aggregating all information from both directions for context words. The BiLSTM model was built according to the following definitions:

Word embedding layer: For $s \in S$, let $W = \{w_1, w_2, \ldots, w_m\}$ be a set of words in given sentence s. For $w_i \in W$, $i = [1, m]$, each word w_i is mapped to a vector, denoted by $x_i \in R^{d_w}$, to obtain a word embeddings matrix $X \in R^{m \times d_w}$, where m is the number of words in sentence s, and d_w is dimension of the word vector. The value of vector x_i is identified as follows:

$$x_i = GloVe(w_i) \tag{5}$$

where $GloVe(w_i)$ is the vector corresponding to word w_i that is extracted from the GloVe embeddings [30].

BiLSTM layer: This layer aims to aggregate the contextual information from both directions for words. BiLSTM includes a forwards LSTM to read the sentence from left to right and a backwards LSTM to read the sentence from right to left. BiLSTM maps each word vector x_i to a pair of hidden vectors $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$. This layer plays the main role in the contextualized word representation step. This layer is formulated as follows:

$$\overrightarrow{h_i} = \overrightarrow{lstm}(x_i), i = [1, m] \tag{6}$$

Table 2
List of commonly notations.

Notation	Description	Notation	Description
T	The set of opinions	S	The set of sentences in an opinion
W	The set of words in a sentence	t	An opinion $t \in T$
G^S	A syntax-based graph	s	A sentence $s \in S$
V^S	The set of nodes on graph G^S	w_i	The i-th word $w_i \in W$
E^S	The set of edges on graph G^S	C	The set of concepts used in graph G^K
A^S	The adjacency matrix of graph G^S	m	The number of words in sentence s
H^S	Node feature matrix of graph G^S	e	The number of sentences
G^K	A syntax-based graph	f	The number of concepts
V^K	The set of nodes on graph G^K	X	Word Embeddings matrix
E^K	The set of edges on graph G^K	v_i	A node on graph G
A^K	The adjacency matrix of graph G^K	v_i^S	A node on graph G^S
H^K	Node feature matrix of graph G^K	v_i^K	A node on graph G^K
G^C	A syntax-based graph	v_i^C	A node on graph G^C
V^C	The set of nodes on graph G^C	$\|V\|$	The number of nodes on graph G
E^C	The set of edges on graph G^C	H	Contextualized word representations
A^C	The adjacency matrix of graph G^C	$\odot$	Element-wise product
H^C	Node feature matrix of graph G^C	d_h	The dimension of hidden vectors
G	An SSC graph	d_w	The dimension of word embeddings
V	The set of nodes on graph G	H^{cv}	Concept vectors
E	The set of edges on graph G	H^{sp}	Sentiment score vector of words
A	The adjacency matrix of graph G	H^{ct}	Context vector of words
Q	Node feature matrix of graph G	$\oplus$	Concatenation operator

$$\overleftarrow{h_i} = \overleftarrow{lstm}(x_i), i = [m, 1] \tag{7}$$

$$h_i = \left[\overrightarrow{h_i}, \overleftarrow{h_i}\right] \tag{8}$$

where $\overrightarrow{lstm}$ and $\overleftarrow{lstm}$ are the forwards LSTM and the backwards LSTM, respectively. $\overrightarrow{h}_i$ and $\overleftarrow{h_i}$ are the hidden state of $\overrightarrow{lstm}$ and the hidden state of $\overleftarrow{lstm}$, respectively; and h_i is the contextualized vector of word w_i. For the i-th word in sentence s, $\overrightarrow{lstm}$ and $\overleftarrow{lstm}$ are calculated as follows:

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

$$\overrightarrow{G_i} = \begin{bmatrix} \overrightarrow{h}_{i-1} \\ x_i \end{bmatrix} \tag{9}$$

$$f_i = sigmoid\left(W^f . \vec{G_i} + b^f\right) \tag{10}$$

$$in_i = sigmoid\left(W^{in} . \vec{G_i} + b^{in}\right) \tag{11}$$

$$o_i = sigmoid\left(W^o . \vec{G_i} + b^o\right) \tag{12}$$

$$c_i = f_i \odot c_{i-1} + in_i \odot tanh\left(W^c . \vec{G_i} + b^c\right) \tag{13}$$

$$\overrightarrow{h_i} = o_i \odot tanh(c_i) \tag{14}$$

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

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

$$\overleftarrow{G_i} = \begin{bmatrix} \overleftarrow{h}_{i+1} \\ x_i \end{bmatrix} \tag{15}$$

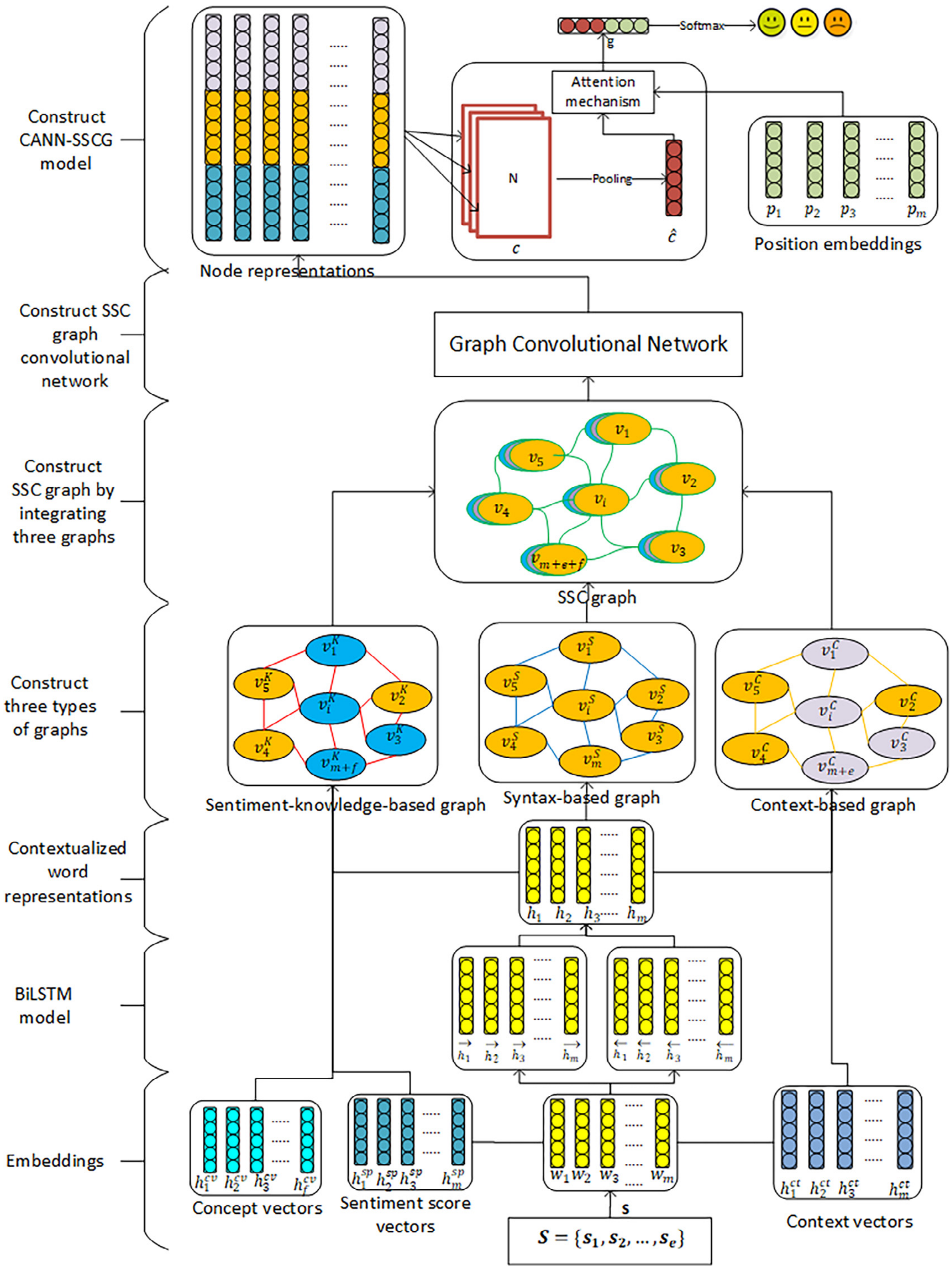


Fig. 1. Overall framework for proposed method.

$$f_i = sigmoid\left(W^f . \overleftarrow{G}_i + b^f\right) \tag{16}$$

$$in_i = sigmoid\left(W^{in} . \overleftarrow{G}_i + b^{in}\right) \tag{17}$$

$$o_i = sigmoid\left(W^o.\overleftarrow{G_i} + b^o\right) \quad (18)$$

$$c_i = f_i \odot c_{i-1} + in_i \odot tanh\left(W^c.\overleftarrow{G_i} + b^c\right) \quad (19)$$

$$\overleftarrow{h_i} = o_i \odot tanh(c_i) \quad (20)$$

where h_{i+1} is the next hidden state of h_i and $h_{m+1} = 0$. f, in, o, c are the operations. $\odot$ is elementwise multiplication. $W^f, W^{in}, W^o, W^c \in R^{d_h \times (d_h + d_w)}, b^f, b^{in}, \ b^o, b^c \in R^{d_h}$ are LSTM parameters; d_h is the dimension of the hidden vectors (representations).

Output layer: Therefore, from the word embedding matrix $X = (x_1, x_2, \ldots, x_m) \in R^{m \times d_w}$, we obtain the contextualized word representations $H = (h_1, h_2, \ldots, h_m) \in R^{m \times d_h}$, where d_h is the dimension of the hidden state vectors.

4.2. Construction of the three subgraphs

4.2.1. Syntax-based graph

The difference from the previous methods, such as [50], is that we consider the syntactic dependency relationships between the word nodes when constructing a syntax-based graph. Syntactic dependency is a relation between two words, where one is the governor word and the other is the dependent word, in a sentence.

Definition 4. A syntax-based graph is denoted by $G^S = \left(V^S, E^S, A^S\right)$, where V^S is a set of nodes corresponding to words in the sentence, E^S is a set of edges containing all pairs of nodes, and $A^S \in R^{m \times m}$ is a syntactic-based adjacency matrix representing the syntactic dependency relations between nodes. G^S has a node feature matrix $H^S = [H] \in R^{m \times d_h}$, where each row H_i^S represents the feature vector of the word node $v_i^S \in V^S$. The elements of the adjacency matrix A^S are determined as follows:

$$A_{ij}^S = \begin{cases} DT\left(v_i^S, v_j^S\right), & \text{if } v_i^S, v_j^S \in W \wedge v_i^S \neq v_j^S \\ 1, & \text{if } v_i^S, v_j^S \in W \wedge v_i^S = v_j^S \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

where $DT\left(v_i^S, v_j^S\right)$ is the syntactic dependency relationship between the nodes [14] v_i^S, v_j^S in the dependency tree. The Stanford dependencies[4] are used to obtain the syntactic dependency tree of a sentence. The relationship $DT\left(v_i^S, v_j^S\right)$ [16] is calculated as follows:

$$DT\left(v_i^S, v_j^S\right) = \frac{1}{m-1} \sum_{s \neq j}^{m-1} d(i, k|j) \quad (22)$$

where

$$d(i, k|j) = C(i, k) - PC(i, k|j) \quad (23)$$

where

$$PC(i, k|j) = \frac{C(i, k) - C(i, j)C(k, j)}{\sqrt{\left[1 - C^2(i, j)\right]\left[1 - C^2(k, j)\right]}} \quad (24)$$

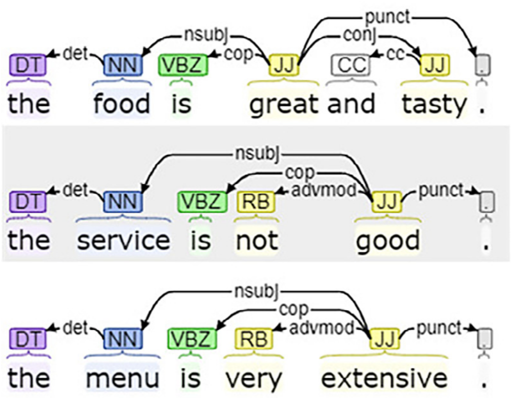
$PC(i, k|j)$ is a statistical measure indicating how a third node affects the correlation between two other nodes. $C(i, k), C(i, j), C(k, j)$ are the node correlations and are calculated as follows:

$$C(i, j) = \frac{\left(\overrightarrow{x_i} - \mu_i\right)\left(\overrightarrow{x_j} - \mu_j\right)}{\sigma_i \sigma_j} \quad (25)$$

where $\overrightarrow{x_i}$ and $\overrightarrow{x_j}$ are the variance value of vectors x_i, x_j (the GloVe vector of words w_i, w_j corresponding to nodes v_i^S, v_j^S). σ_i and σ_j are the standard deviation of vectors x_i and x_j. μ_i and μ_j are the mean value of vectors x_i and x_j.

Fig. 2 shows the dependency tree and the adjacency matrix of the syntax-based graph for the opinion *"The food is great and tasty. The service is not good. The menu is very extensive."*

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



	the	food	is	great	and	tasty	...	service	is	not	...
the	1	DT	0	0	0	0	...	0	0	0	...
food	DT	1	0	DT	0	0	...	0	0	0	...
is	0	0	1	DT	0	0	...	0	0	0	...
great	0	DT	DT	1	0	DT	...	0	0	0	...
and	0	0	0	0	1	DT	...	0	0	0	...
tasty	0	0	0	DT	DT	1	...	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
service	0	0	0	0	0	0	...	1	0	0	...
is	0	0	0	0	0	0	...	0	1	0	...
not	0	0	0	0	0	0	...	0	0	1	...
...	...	...	...	...	...	...	...	...	...	...	...

Fig. 2. An example of dependency tree and the adjacency matrix of the syntax-based graph.

4.2.2. Semantic-based graph

This graph represents an opinion based on the word-concept relations, concept-concept relations, and sentiment scores of words. It is built based on improving the previous method [50] by considering the co-occurrence relations of words and concepts. To calculate the co-occurrence information of words and concepts, we use a fixed size sliding window on all opinions to determine their co-occurrence.

Let $C = \{c_1, c_2, \ldots, c_f\}$ be a set of concepts including mood tags and semantically similar words of words appearing in sentences s (see the example in Fig. 3). These concepts are a common sense knowledge base for sentiment analysis that is created based on SenticNet 6 introduced by Cambria et al. [4] via an API.[5]

Definition 5. A semantic-based graph is denoted by $G^K = \left(V^K, E^K, A^K\right)$, where V^K is a set of nodes corresponding to the words and concepts in a sentence, E^K is a set of edges containing all pairs of nodes, and $A^K \in R^{(m+f)\times(m+f)}$ is a semantic-based adjacency matrix representing the semantic relations between nodes. G^K has a node feature matrix $H^K = [H^{cv} \oplus H^{sp} \oplus H] \in R^{(m+f)\times d_h}$, where each row represents the feature vector of the word or concept node v_i^K, and H^{cv} is a concept vector generated based on common sense knowledge embeddings [3,23]. H^{sp} is a sentiment score vector of words and is created as follows:

$$H_i^{sp} = [score(w_i)], w_i \in W \cup C \tag{26}$$

where

$$score(w_i) = \begin{cases} Sc_i, & \text{if } w_i \text{ is a clear sentiment phrase} \\ Sf_i, & \text{if } w_i \text{ is a fuzzy sentiment phrase} \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

where $\oplus$ is the vector concatenation operator. The way to calculate the value of parameters Sc_i and Sf_i is presented in detail in [32,33]. The elements of the adjacency matrix A^K are determined as follows:

$$A_{ij}^K = \begin{cases} PMI\left(v_i^K, v_j^K\right), & \text{if } v_i^K, v_j^K \in C \wedge PMI(.) > 0 \wedge v_i^K \neq v_j^K \\ 1, & \text{if } v_i^K \in W \wedge v_j^K \in C \\ 1, & \text{if } v_i^K = v_j^K \\ 0, & otherwise \end{cases} \tag{28}$$

where $PMI\left(v_i^K, v_j^K\right)$ is a popular measure to calculate the weights between two nodes of words based on word co-occurrence information. This measure obtains better accuracy than using the word co-occurrence count [49]. The value of $PMI\left(v_i^K, v_j^K\right)$ is identified as follows:

[5] https://sentic.net/api/.

```
sn = SenticNet()

sn.moodtags('food')
['#joy', '#joy']

sn.semantics('food')
['beverage', 'health', 'wealth', 'shelter', 'collectable']

sn.moodtags('great')
['#joy', '#eagerness']

sn.semantics('great')
['best', 'excellent', 'well_done', 'goodness', 'good_quality']

sn.moodtags('tasty')
['#joy', '#eagerness']

sn.semantics('tasty')
['delicious', 'yummy', 'sweet', 'creamy', 'high_calorie']
```

	the	food	is	great	and	tasty	wealth	health	...	best	goodness	...	sweet	creamy	...	#joy	#eagerness
the	1	0	0	0	0	0	0	0	...	0	0	...	0	0	...	0	0
food	0	1	0	0	0	0	1	1	...	0	0	...	0	0	...	1	0
is	0	0	1	0	0	0	0	0	...	0	0	...	0	0	...	0	0
great	0	0	0	1	0	0	0	0	...	1	1	...	0	0	...	1	1
and	0	0	0	0	1	0	0	0	...	0	0	...	0	0	...	0	0
tasty	0	0	0	0	0	1	0	0	...	0	0	...	1	1	...	1	1
wealth	0	1	0	0	0	0	1	PMI	...	0	0	...	0	0	...	PMI	0
health	0	1	0	0	0	0	PMI	1	...	0	0	...	0	0	...	PMI	0
...	...	...	...	...	...	...	...	...	...		...	...	...	...	...	...	...
best	0	0	0	1	0	0	0	0	...	1	PMI	...	0	0	...	PMI	PMI
goodness	0	0	0	1	0	0	0	0	...	PMI	1	...	0	0	...	PMI	PMI
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
sweet	0	0	0	0	0	1	0	0	...	0	0	...	1	PMI	...	PMI	PMI
creamy	0	0	0	0	0	1	0	0	...	0	0	...	PMI	1	...	PMI	PMI
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
#joy	0	1	0	1	0	1	PMI	PMI	...	PMI	PMI	...	PMI	PMI	...	1	PMI
#eagerness	0	0	0	1	0	1	0	0	...	PMI	PMI	...	0	0	...	PMI	1

Fig. 3. An example of the adjacency matrix of the semantic-based graph.

$$PMI\left(v_i^K, v_j^K\right) = log\frac{p(i,j)}{p(i)p(j)} \tag{29}$$

$$p(i,j) = \frac{|D(i,j)|}{|D|} \tag{30}$$

$$p(i) = \frac{|D(i)|}{|D|} \tag{31}$$

$$p(j) = \frac{|D(j)|}{|D|} \tag{32}$$

where $|D|$ is the total number of words and concepts in T. $|D(i)|$ and $|D(j)|$ are the number of occurrences in T of the token corresponding to v_i^K and v_j^K, respectively. $|D(i,j)|$ is the number of co-occurrences in T of both tokens corresponding to v_i^K and v_j^K.

Fig. 3 shows the creation steps of the set of knowledge concepts and the adjacency matrix of the semantic-based graph for the opinion *"The food is great and tasty. The service is not good. The menu is very extensive."*

4.2.3. Context-based graph

This graph represents an opinion based on word co-occurrence, sentence-word relations, and relations between sentences. It is built based on improving the TextGCN in [49] by considering the relations between sentences.

Definition 6. A context-based graph is denoted by $G^C = \left(V^C, E^C, A^C\right)$, where V^C is a set of nodes corresponding to the words and sentences in an opinion, E^C is a set of edges containing all pairs of nodes, and $A^C \in R^{(m+e)\times(m+e)})$ is a context-based adjacency matrix representing the context relations between nodes. G^C has a node feature matrix $H^C = \left[H^{ct} \oplus H\right] \in R^{(m+e)\times d_h}$, where each row H_i^C represents the feature vector of the word or sentence node v_i^C, and H^{ct} is the context vector of words identified based on the context2vec model[6] [5]. The elements of the adjacency matrix A^C are determined as follows:

$$A_{ij}^C = \begin{cases} COS\left(v_i^C, v_j^C\right), & \text{if } v_i^C, v_j^C \in S \wedge v_i^C \neq v_j^C \\ TF-IDF_{ij}, & \text{if } v_i^C \in S \wedge v_j^C \in W \\ 1, & \text{if } v_i^C = v_j^C \\ 0, & otherwise \end{cases} \tag{33}$$

[6] https://github.com/senticnet/context2vec.

where $COS\left(v_i^C, v_j^C\right)$ is the distance between two sentences s_i and s_j. S is the set of sentences in an opinion. W is the set of words. The value of $COS\left(v_i^C, v_j^C\right)$ is calculated as follows:

$$COS\left(v_i^C, v_j^C\right) = \frac{COS\left(\vec{s_i}, \vec{s_j}\right)}{|i-j|} \tag{34}$$

where $\vec{s_i}$ and $\vec{s_j}$ are the vectors of sentences s_i and s_j, respectively.

$$\vec{s_i} = \frac{1}{m_1}\sum_{i=1}^{m_1} x_i, w_i \in W \tag{35}$$

$$\vec{s_j} = \frac{1}{m_2}\sum_{j=1}^{m_2} x_j, w_j \in W \tag{36}$$

where m_1 and m_2 are the number of words in sentences s_i and s_j, respectively.

Fig. 4 shows the set of sentences in the opinion and the adjacency matrix of the context-based graph for the opinion "*The food is great and tasty. The service is not good. The menu is very extensive.*"

At the end of this step, we finish answering the first research question presented in Section 3.3.

4.3. SSC graph construction

In this section, we integrate the information of the syntactic graph, the semantic graph, and the context graph into a new heterogeneous graph called the syntactic, semantic, and context graph to learn the essential information from the previous three graphs for sentence representation.

Definition 7. An SSC graph, denoted by G, is the integrated graph obtained by combining the information of syntax-based graphs, semantic-based graphs, and context-based graphs. $G = (V, E, A)$ is defined by a set of nodes $V = V^S \cup V^K \cup V^C$ and a set of edges $E = E^S \cup E^K \cup E^C$. The number of nodes is the total number of words, concepts, and sentences in the opinion. G has a node feature matrix $Q = \left[H^S \oplus H^K \oplus H^C\right] \in R^{|V| \times d_h}$. The adjacency matrix $A \in R^{|V| \times |V|}$ represents the relations between nodes. The elements of the adjacency A are determined as follows:

$$A_{ij} = \begin{cases} DT(v_i, v_j), & \text{if } v_i, v_j \in W \wedge v_i \neq v_j \\ COS(v_i, v_j), & \text{if } v_i, v_j \in S \wedge v_i \neq v_j \\ TF-IDF_{ij}, & \text{if } v_i \in S \wedge v_j \in W \\ PMI(v_i, v_j), & \text{if } v_i, v_j \in C \wedge v_i \neq v_j \\ 1, & \text{if } v_i \in W \wedge v_j \in C \\ 1, & \text{if } v_i = v_j \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

Overall, the SSC graph is a graph with properties, such as the following: (i) heterogeneous; (ii) undirected; (iii) no binary weight; and (iv) each weight type expresses one type of meaning, such as DT-weighted word-word edges capture within-sentence syntactic, COS-weighted sentence-sentence edges (which can span across opinions) capture across-opinion contexts, $TF-IDF$-weighted sentence-word edges capture within-sentence context, and PMI-weighted concept-concept edges (which can span across sentences) capture across-sentence sentiment-knowledge.

Fig. 5 shows the adjacency matrix of the SSC graph for the opinion "*The food is great and tasty. The service is not good. The menu is very extensive.*"

At the end of this step, we finish answering the second research question presented in Section 3.3.

4.4. CANN-SSCG model

The CNN model was successfully applied by Kim et al. [17] for sentence classification. It has become a significant deep learning model used in the natural language processing field since the study by Collobert et al. [6], who applied CNN successfully in SA. The CANN model [36] is an extension of the CNN model based on the attention mechanism. The CANN model was built in the following phases:

SSC node embeddings layer: Each node v_i on the SSC graph is represented by a vector $h_i \in R^{d_h}$. The entire graph obtains a nodes feature matrix $Q \in R^{|V| \times d_h}$ and an adjacency matrix $A \in R^{|V| \times |V|}$. After building the SSC graph, we feed matrices Q and A into a simple two-layer GCN proposed by [19] as follows:

	the	food	is	great	and	tasty	...	s1	s2	s3
the	1	0	0	0	0	0	...	TF	TF	TF
food	0	1	0	0	0	0	...	TF	0	0
is	0	0	1	0	0	0	...	TF	TF	TF
great	0	0	0	1	0	0	...	TF	0	0
and	0	0	0	0	1	0	...	0	0	0
tasty	0	0	0	0	0	1	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...
s1	TF	TF	TF	TF	TF	TF	...	1	COS	COS
s2	TF	0	TF	0	0	0	...	COS	1	COS
s3	TF	0	TF	0	0	0		COS	COS	1

Fig. 4. An example of the adjacency matrix of the context-based graph.

	the	food	is	great	and	tasty	wealth	health	...	best	goodness	...	sweet	creamy	...	#joy	#eagerness	...	s1	s2	s3
the	1	DT	0	0	0	0	0	0	...	0	0	...	0	0	...	0	0	...	TF	TF	TF
food	DT	1	0	DT	0	0	1	1	...	0	0	...	0	0	...	1	0	...	TF	0	0
is	0	0	1	DT	0	0	0	0	...	0	0	...	0	0	...	0	0	...	TF	TF	TF
great	0	DT	DT	1	0	DT	0	0	...	1	1	...	0	0	...	1	1	...	TF	0	0
and	0	0	0	0	1	DT	DT	0	...	0	0	...	0	0	...	0	0	...	0	0	0
tasty	0	0	0	DT	DT	1	DT	0	...	0	0	...	1	1	...	1	1	...	0	0	0
wealth	0	1	0	0	0	0	1	PMI	...	0	0	...	0	0	...	PMI	0	...	0	0	0
health	0	1	0	0	0	0	PMI	1	...	0	0	...	0	0	...	PMI	0	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
best	0	0	0	1	0	0	0	0	...	1	PMI	...	0	0	...	PMI	PMI	...	0	0	0
goodness	0	0	0	1	0	0	0	0	...	PMI	1	...	0	0	...	PMI	PMI	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	0	0	0
sweet	0	0	0	0	0	1	0	0	...	0	0	...	1	PMI	...	PMI	PMI	...	0	0	0
creamy	0	0	0	0	0	1	0	0	...	0	0	...	PMI	1	...	PMI	PMI	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
#joy	0	1	0	1	0	1	PMI	PMI	...	PMI	PMI	...	PMI	PMI	...	1	PMI	...	0	0	0
#eagerness	0	0	0	1	0	1	0	0	...	PMI	PMI	...	0	0	...	PMI	1	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
s1	TF	TF	TF	TF	TF	TF	0	0	...	0	0	...	0	0	...	0	0	...	1	COS	COS
s2	TF	0	TF	0	0	0	0	0	...	0	0	...	0	0	...	0	0	...	COS	1	COS
s3	TF	0	TF	0	0	0	0	0	...	0	0	...	0	0	...	0	0	...	COS	COS	1

Fig. 5. The adjacency matrix of the SSC graph.

$$H^1 = ReLU\left(M \cdot H^0 \cdot W^1 + b^1\right) \tag{38}$$

Hence,

$$H^2 = ReLU\left(M \cdot H^1 \cdot W^2 + b^2\right) \tag{39}$$

That means:

$$X = H^2 = ReLU\Big(ReLU\Big(M \cdot H^0 \cdot W^1 + b^1\Big) \cdot W^2 + b^2\Big)\Big) \quad (40)$$

where $X \in R^{|V| \times d_h}$, and $H^0 = Q$. $ReLU$ is a rectified linear function. $W^1 \in R^{d_h \times |V|}$ and $W^2 \in R^{|V| \times d_h}$ are the weight matrices created for the i-th layer. b^1 and b^2 are the biases of the two layers.

$$M = D^{-0.5} A D^{-0.5} \quad (41)$$

M is the normalized symmetric adjacency matrix of $A; D$ is the degree matrix of A, where:

$$D_{ii} = \sum_j A_{ij} \quad (42)$$

Convolutional layer: The main task of this layer is to create a feature map, denoted by c, from the SSC node embeddings layer. Parameter c is created based on a filter $N \in R^{q \times d_h}$ of length q from i to $i + q - 1$ to slide and extract important information. Each time filter N is slid, a new feature vector is created as follows:

$$c_i = ReLU\big(N \ominus X_{i:i+|V|-1} + b\big) \quad (43)$$

where i represents the order of the rows (node vectors) and $i = [1, |V|]$, $\ominus$ is the convolution operation, and $ReLU$ is a rectified linear unit function. b is a bias term. X_i is the representation of node v_i on the SSC graph obtained by using a GCN. Therefore, from a set of nodes, the features map is created as follows:

$$c = \big[c_1, c_2, .., c_{|V|}\big] \quad (44)$$

Max-pooling layer: Because the dimension of feature map c is dependent on the size of both matrices X and N, in other words, the dimension of feature vectors $c_i \in c$ will differ if we have sentences of different lengths and filters of different sizes. Therefore, the max pooling layer aims to create a new feature map, denoted by $\hat{c}$, with the same size feature vectors by extracting the largest number from each c_i vector. After this step, we obtain a new feature map as follows:

$$\hat{c} = \big[\hat{c}_1, \hat{c}_2, .., \hat{c}_{|V|}\big] \quad (45)$$

where $\hat{c}_i = Max(c_i)$. Then vector $\hat{c}$ is fed into next layer.

Position embeddings: This step aims to determine the position information of the words indicating the aspects in the sentence (called aspect words). Assuming that we have aspect word $w_j, j = [1, k]$, where k is the number of aspects in the given opinions, the position embeddings of words w_i in sentence s, denoted by $p_i \in R^k$, are created by considering the following steps. First, we determine the relative distance of word w_i to aspect words w_j. Finally, vector p_i is created based on these relative distances as follows:

$$p_i = [d_{i1}, d_{i2}, \ldots, d_{ik}] \in R^k \quad (46)$$

where d_{ij} is the distance from word w_i to aspect word w_j. In addition, the value of d_{ij} is calculated as follows:

$$d_{ij} = \frac{|position(w_i) - position(w_j)|}{|max_{s \in S} length(s)|} \quad (47)$$

where w_j represents a word indicating an aspect and $position(w_j) = 0$. w_i indicates a word not expressing an aspect, and the position of word w_i is calculated by the distance to aspect word w_j in the sentence as $|i - j|$. $|max_{s \in S} length(s)|$ is a function to give the number of words of the longest sentence in the given opinion.

Attention mechanism: Traditional CNN methods often use min/max/average pooling as the attention layer. However, not all word nodes are essential for representing the meaning of sentences. Thus, the attention mechanism [38,44] is proposed to extract the critical word nodes to identify aspects and their sentiments. In this study, the attention mechanism is used to integrate the positions of words indicating aspects and aggregate the representations of these informative words. This technique can highlight the differences in the words indicating aspect-level sentiment without expressing aspect-level sentiment. The position attention mechanism is applied to enhance the discrimination between valuable and useless features regarding aspect sentiment.

First, a score function f is defined as the following equation to capture the importance of the information in the sentence.

$$f(\hat{c}_i, p_i) = tanh(F \cdot [\hat{c}_i \oplus p_i] + b) \quad (48)$$

where $F \in R^{(d_h+k) \times (d_h+k)}$ is the weight matrix. $\oplus$ is the concatenation operator between two vectors. $b \in R^{d_h \times k}$.

Then, the attention vector is generated as follows:

$$g = \sum_{i=1}^{|V|} \alpha_i \hat{c}_i \quad (49)$$

where

$$\alpha_i = \frac{exp(f(\hat{c}_i, p_i))}{\sum_{j=1}^{|V|} exp(f(\hat{c}_j, p_j))} \quad (50)$$

Fully connected layer: This layer is used to fine-tune the sentiment characteristic of the previous layer and classify aspect-level sentiment using the softmax function as follows:

$$\hat{y} = Softmax(E \cdot g + b) \quad (51)$$

where $E \in R^{l \times |V|}$ is a weight matrix of the softmax function. $b \in R^l$ is a bias of the softmax function. l is the number of classifications.

Model training: The CANN-SSCG model is trained by minimizing the cross-entropy error of the predicted and true label distributions as the following equation:

$$L = -\sum_i^l y_i log(\hat{y}_i) + \lambda \|\theta\|^2 \quad (52)$$

where l represents the number of classifications in the training set, y_i indicates the real label matrix of the i-th class, and $\hat{y}_i$ represents the predicted probability for the i-th class. λ is the coefficient of L_2 regulation. θ is the parameter set from the convolutional layer, max-pooling layer, attention layer, and fully connected layer.

At the end of this step, we finish answering the final research question presented in Section 3.3.

5. Experiment

5.1. Data acquisition

For the experiments, we used English sentences from three datasets, including the SemEval'14 dataset[7] introduced in [9,34], tweets[8] presented by Dong et al. [8], and the MAMS dataset[9] represented in [15,35]. Multiaspect multisentiment (MAMS) is a dataset proposed by Jiang et al. [15] to increase the challenge in aspect-level SA to promote this field's development. Each sentence in this dataset includes at least two aspects with different sentiments. The restaurant dataset is given by Ganu et al. [9] and contains reviews related to restaurants. Each sentence in the restaurant dataset included at least one aspect and aspect sentiment. The tweet dataset consists of related celebrities, products, and companies that are posted on Twitter. Each sentence included at least one aspect and sentiment. Since the sentences in all datasets include acronyms, spelling errors, and symbols, it was necessary to correct them. First, we fixed these spellings using the Python-based Aspell library.[10] Then, for all datasets, the words indicating an aspect were labelled *aspect*, and the words indicating an aspect sentiment were assigned one of the three labels: *positive*, *neutral*, and *negative*. The details of these datasets are presented in Table 3.

5.2. Experimental setup

The proposed model was built using Python 3.7, PyTorch 1.1.0, Scikit-learn 0.21.3, SciPy 1.3.1, NumPy 1.16.2, NetworkX 2.4, and TensorFlow 2.1.0. The experiments were run using an 8 GB GeForce RTX 2070 on a computer with an Intel Core-i5 7th generation CPU and 32 GB of RAM.

The codes of the model are based on NetworkX,[11] the GCN in PyTorch version,[12] and Text-GCN.[13] Furthermore, our proposal was implemented following mathematical models with a list of hyperparameter values as follows. As only the MAMS dataset provides the development data, we fine-tuned the model's hyperparameters on the development data of the MAMS and used the same hyperparameters for the other datasets. The following hyperparameters were suggested for the proposed model. In all experiments, we used 300-dimensional word vectors pretrained by GloVe [30] to initialize the word embedding vectors. The dimension of the hidden states in the BiLSTM model was set to 100. In addition, the number of layers of the GCN model was two, and the dimension of the hidden layer was 100, thus yielding the same result as Kipf and Welling [19]. For the CANN algorithm, the hyperparameters were set as follows: the filter size was 3 or 4, the number of filters was 100, and the dimension of the hidden layer was 100. All weight matrices were initialized randomly using the uniform distribution U(−0.1,0.1), and biases were set as 0. The Adam optimizer [18], with a learning rate of 0.001, was used to train our models; and the dropout rate was set to 0.5 to avoid overfitting. The experimental results showed that smaller or larger values of hyperparameters did not lead to better performance. All models were run five times, and the average results for the test datasets were reported. The hyperparameters were fine-tuned for all baselines based on the cross-validation technique. L_2 norm regularization was set as 10^{-5}.

[7] https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools.
[8] http://goo.gl/5Enpu7.
[9] https://github.com/siat-nlp/MAMS-for-ABSA.
[10] https://pypi.org/project/aspell-python-py2/.
[11] https://github.com/networkx/networkx.
[12] https://github.com/tkipf/pygcn.
[13] https://github.com/yao8839836/text_gcn.

Table 3
Statistics of datasets.

Dataset	Training set			Testing set			Developing set		
	#Positive	#Neutral	#Negative	#Positive	#Neutral	#Negative	#Positive	#Neutral	#Negative
Tweet	1562	3124	1562	173	346	173	–	–	–
Restaurant	2164	637	807	728	196	196	–	–	–
MAMS	3380	5042	2764	400	607	329	403	604	325

Table 4
Comparison performance of methods for Tweet dataset (%).

Aspect-term SA with Word2Vec embeddings									
Method	Positive			Neutral			Negative		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	72.36	71.84	72.1	72.05	71.52	71.78	70.62	71.04	70.83
FEM+BiLSTM+CRF	72.31	72.81	72.56	72.61	71.59	72.10	71.26	70.71	70.98
BANN	73.26	71.92	72.58	72.36	73.27	72.81	73.42	72.05	72.73
CNN-based	71.09	70.52	70.80	70.81	71.19	71.00	71.82	70.91	71.36
CANN	72.79	72.63	72.71	71.48	70.29	70.88	71.52	72.08	71.80
CANN-SG	72.37	73.02	72.69	73.21	72.72	72.96	73.64	74.20	73.92
CANN-SKG	71.30	70.69	70.99	73.06	73.83	73.44	72.01	74.19	73.08
CANN-CG	72.12	72.64	72.38	74.06	72.04	73.04	72.28	75.21	73.72
CANN-SSG	74.63	73.09	73.85	74.92	75.05	74.98	75.13	73.62	74.37
CANN-SCG	74.42	75.25	74.83	75.16	74.05	74.60	74.56	76.01	75.28
CANN-SKCG	75.25	73.05	74.13	74.95	74.08	74.51	75.65	75.38	75.51
CANN-SSCG	76.81	77.75	77.28	77.06	77.31	77.18	77.09	78.92	77.99
Aspect-term SA with GloVe embeddings									
Method	Positive			Neutral			Negative		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	71.27	72.38	71.82	71.01	70.94	70.97	72.02	71.25	71.63
FEM+BiLSTM+CRF	72.97	71.28	72.12	72.07	70.75	71.40	72.26	72.25	72.25
BANN	74.33	73.95	74.14	74.52	74.50	74.51	72.67	72.52	72.59
CNN-based	71.10	69.95	70.52	70.81	71.45	71.13	71.08	69.19	70.12
CANN	70.06	71.91	70.97	71.25	72.49	71.86	72.31	74.05	73.17
CANN-SG	75.03	74.57	74.80	75.03	75.46	75.24	74.11	73.82	73.96
CANN-SKG	74.04	75.82	74.92	74.06	74.15	74.10	75.21	75.38	75.29
CANN-CG	74.21	74.09	74.15	74.35	74.37	74.36	75.03	76.16	75.59
CANN-SSG	75.92	75.23	75.57	75.91	74.41	75.15	77.32	76.28	76.80
CANN-SCG	75.14	77.09	76.10	78.87	77.01	77.93	78.20	77.53	77.86
CANN-SKCG	76.05	76.96	76.50	78.42	76.09	77.24	79.15	77.02	78.07
CANN-SSCG	78.64	79.05	78.84	79.04	78.41	78.72	79.03	80.19	79.61

5.3. Baselines

In this study, we divided baselines into two small groups: baseline methods and ablated methods. The baselines were constructed as follows:

- Baseline methods
  - BiLSTM+CRF: A method for aspect-level SA based on a combination of BiLSTM and conditional random field (CRF) models [31].
  - Feature Ensemble Model (FEM) + BiLSTM + CRF: Incorporates the improvement of word representations into a combination of BiLSTM and CRF models [32].
  - BANN: Interactive attention neural network for aspect-level SA using the BiLSTM algorithm [36].
  - CNN-based method: Improves attention-based input layers to consider aspects. Subsequently, improved input layers are integrated into a CNN model to introduce contextual words [46].
  - CANN: The interactive attention neural network for aspect-level SA using a CNN algorithm [36].
- Ablated models
  - CANN-SG applies the CANN algorithm to only the syntax-based graph.
  - CANN-SKG applies the CANN algorithm to only the semantic-based graph.
  - CANN-CG applies the CANN algorithm to only the context-based graph.

– CANN-SSG applies the CANN algorithm to the combination of the syntax-based graph and the semantic-based graph.
– CANN-SCG applies the CANN algorithm to the combination of the syntax-based graph and the context-based graph.
– CANN-SKCG applies the CANN algorithm to the combination of the semantic-based graph and the context-based graph.
– CANN-SSCG fully implements our proposal.

5.4. Experimental result and discussion

Given a set of opinion sentences, the aspect-term extraction task seeks to identify all aspect terms present in each sentence (e.g., *"fish"*, *"soup"*, *"appetizer"*, *"price"*, *"waiter"*). Given a set of aspect terms (this is the result of the aspect prediction task), the aspect-term SA task determines the polarity of each aspect term (e.g., *"positive"*, *"negative"*, *or "neutral"*). In this study, we focus on the aspect-term SA. Therefore, we only evaluate the performance of our proposal in terms of aspect-term SA. In addition, to understand the justification when choosing the GloVe as word embeddings in the proposed model, we implemented the methods with both word2vec and GloVe embeddings.

The results of the aspect-term SA are shown in Tables 4-7. Each table includes two main parts. The upper part presents the performance of methods for the aspect-term SA with word2vec embeddings (it is abbreviated as aspect-term SA with word2vec embeddings) while the lower part presents the efficacy of methods for tasks with GloVe embeddings (it is abbreviated as aspect-term SA with GloVe embeddings). In both parts, each line lists the *Precision*, *Recall*, and F_1*score* of each method on a specific dataset, where the best scores are in bold.

Tables 4–7 present the performance of the models on the test sets of the three datasets. From these tables, we illustrate the performance in Figs. 6–8. Then, we present some observations in Table 8.

Based on the observations, we summarize the following conclusions:

- Although the number of samples for the training of the tweet dataset is two times higher than that of the restaurant dataset, most methods still achieve high performance for the restaurant dataset, except for BiLSTM+CRF, the CNN-based method, and FEM+BiLSTM+CRF. This is because although the restaurant dataset consists of fewer samples, it focuses on expressing the users' sentiment regarding aspects of only one topic (a restaurant). The tweet dataset samples, how-

Table 5
Comparison performance of methods for Restaurant dataset (%).

Aspect-term SA with Word2Vec embeddings									
Method	Positive			Neutral			Negative		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	70.89	69.42	70.15	70.92	70.23	70.57	69.72	71.21	70.46
FEM+BiLSTM+CRF	71.08	70.10	70.59	71.18	72.90	72.03	72.21	71.02	71.61
BANN	74.02	72.38	73.19	73.72	74.09	73.90	74.06	73.28	73.67
CNN-based	70.42	71.25	70.83	71.18	69.91	70.54	70.28	70.19	70.23
CANN	72.56	72.81	72.68	73.65	72.89	73.27	72.98	73.65	73.31
CANN-SG	74.92	75.02	74.97	74.58	73.06	73.81	75.24	74.09	74.66
CANN-SKG	74.38	74.02	74.20	75.93	74.25	75.08	73.98	75.73	74.84
CANN-CG	73.89	74.23	74.06	74.09	73.58	73.83	74.38	74.94	74.66
CANN-SSG	75.09	73.65	74.36	75.05	76.41	75.72	76.93	75.06	75.98
CANN-SCG	74.15	76.53	75.32	76.04	75.75	75.89	75.06	75.95	75.50
CANN-SKCG	75.76	74.16	74.95	76.19	76.81	76.50	76.28	74.52	75.39
CANN-SSCG	78.09	77.18	77.63	78.95	79.03	78.99	77.85	79.42	78.63
Aspect-term SA with GloVe embeddings									
Method	Positive			Neutral			Negative		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	71.32	70.61	70.96	71.53	70.73	71.13	70.82	71.15	70.98
FEM+BiLSTM+CRF	71.14	72.05	71.59	72.34	71.17	71.75	70.76	71.48	71.12
BANN	74.63	75.51	75.07	75.32	74.90	75.11	73.28	72.75	73.01
CNN-based	70.27	69.38	69.82	71.01	69.94	70.47	71.02	71.25	71.13
CANN	72.81	72.36	72.58	72.82	72.70	72.76	72.65	73.85	73.25
CANN-SG	73.98	74.80	74.39	75.79	74.80	75.29	75.03	73.06	74.03
CANN-SKG	74.71	75.92	75.31	73.70	75.10	74.39	75.29	76.06	75.67
CANN-CG	73.66	75.86	74.74	74.68	75.03	74.85	75.71	75.78	75.74
CANN-SSG	76.81	74.85	75.82	75.29	75.41	75.35	77.75	76.04	76.89
CANN-SCG	75.59	76.95	76.26	79.96	76.88	78.39	76.63	79.43	78.00
CANN-SKCG	76.18	77.61	76.89	78.81	78.95	78.88	78.97	77.62	78.29
CANN-SSCG	79.79	79.57	79.68	79.03	79.12	79.07	79.94	81.01	80.47

Table 6
Comparison performance of methods for MAMS dataset (%).

Aspect-term SA with Word2Vec embeddings									
Method	Positive			Neutral			Negative		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	58.81	59.46	59.13	56.71	57.19	56.95	59.21	57.27	58.22
FEM+BiLSTM+CRF	58.63	59.22	58.92	58.57	59.76	59.16	58.44	59.19	58.81
BANN	58.62	60.96	59.77	59.02	60.39	59.70	60.38	59.01	59.69
CNN-based	57.45	58.72	58.08	57.18	54.91	56.02	56.28	58.19	57.22
CANN	59.17	58.55	58.86	59.58	58.62	59.10	59.89	58.58	59.23
CANN-SG	60.31	61.09	60.70	59.22	60.15	59.68	59.42	61.31	60.35
CANN-SKG	60.45	60.91	60.68	59.64	60.20	59.92	59.77	60.71	60.24
CANN-CG	60.08	61.12	60.60	60.36	59.51	59.93	60.51	59.82	60.16
CANN-SSG	61.60	60.58	61.09	61.17	60.94	61.05	61.68	61.26	61.47
CANN-SCG	61.01	61.65	61.33	62.49	60.22	61.33	60.36	62.37	61.35
CANN-SKCG	61.26	61.02	61.14	60.56	61.29	60.92	60.76	61.46	61.11
CANN-SSCG	61.92	62.06	61.99	62.31	62.18	62.24	61.16	62.51	61.83
Aspect-term SA with GloVe embeddings									
Method	Positive			Neutral			Negative		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	60.94	59.03	59.97	58.78	57.02	57.89	59.35	59.04	59.19
FEM+BiLSTM+CRF	58.52	59.15	58.83	57.01	59.72	58.33	58.99	59.59	59.29
BANN	59.61	58.09	58.84	59.24	60.28	59.76	60.34	59.15	59.74
CNN-based	56.42	57.25	56.83	57.18	55.91	56.54	57.28	58.19	57.73
CANN	59.90	59.94	59.92	60.10	59.71	59.90	59.27	59.04	59.15
CANN-SG	59.73	60.02	59.87	60.82	59.07	59.93	60.73	59.68	60.20
CANN-SKG	59.29	58.56	58.92	59.46	61.98	60.69	62.29	61.83	62.06
CANN-CG	60.46	60.01	60.23	60.77	59.04	59.89	60.42	62.03	61.21
CANN-SSG	61.89	62.68	62.28	61.12	62.39	61.75	62.89	61.28	62.07
CANN-SCG	61.74	61.43	61.58	62.66	59.91	61.25	60.82	62.79	61.79
CANN-SKCG	60.78	61.76	61.27	61.54	60.92	61.23	61.78	61.13	61.45
CANN-SSCG	62.04	62.85	62.44	62.63	62.05	62.34	62.83	62.90	62.86

ever, express users' sentiments towards various topics (e.g., celebrities, products, and companies). We refer to this as the dispersion of labels. However, why are the BiLSTM+CRF, FEM+BiLSTM+CRF, and CNN-based methods unaffected by this? These methods were proposed based on the tweet dataset; therefore, when extracting features related to aspect sentiment in tweets, the authors may consider collecting features that conform to the tweets.

- Although the tweet and MAMS datasets include aspects belonging to various topics, most methods obtained high performance on the tweet dataset. This result is in contrast with our first conclusion. Per our assessment, this could be because although the aspects belong to various topics, users' sentiment towards these aspects are only assigned one of three labels: *positive, negative, and neutral*. Therefore, the training set contains an increasing number of samples, and better performance is certainly achieved. We refer to this as the concentration of labels in the training data.
- Most methods achieved competitive results on the restaurant dataset but performed poorly on the MAMS dataset. Specifically, for the aspect-term SA with word2vec embeddings, the difference in the F_1*score* of the restaurant dataset and the tweet dataset is at least 0.47% for the FEM+BiLSTM+CRF model and up to 2.21% for the CANN-SKG model. Furthermore, the difference between the tweet and the MAMS datasets is at least 12.22% via the CANN-SKG model and up to 15.46% for the CANN-SSCG model. Additionally, the difference in the F_1*score* of the restaurant dataset and the MAMS dataset is at least 12.29% for the BiLSTM+CRF model and up to 16.02% for the CANN-SSCG model. This is similar to the aspect-term SA with GloVe embeddings. The results show that the difference in the F_1*score* between the models when using the MAMS dataset and the two remaining datasets is somewhat significant. This may be because the MAMS dataset can include more challenges than the restaurant and tweet datasets [15].
- Under the same conditions, the LSTM-based model can achieve better performance than those not based on LSTM (such as the CANN, BANN, CNN-based and BiLSTM+CRF methods). In the case of the CANN and BANN, for aspect-term SA with word2vec embeddings, the BANN model can improved the F_1*score* of the CANN by at least 0.5% and by up to 0.91% for the restaurant and the tweet datasets, respectively. However, when using GloVe embeddings, the BANN model can improve the F_1*score* of the CANN by at least 1.54% and by up to 1.7% for the restaurant dataset and the tweet dataset, respectively. However, the efficacy of the BANN model is lower than that of the CANN model by 0.21% for the MAMS dataset. In the case of BiLSTM+CRF and the CNN-based model, for aspect-term SA with word2vec embeddings, the BiLSTM +CRF model can improve the F_1*score* of the CNN-based model by at least 0.52% and by up to 0.99% for the tweet and the MAMS datasets, respectively. However, its efficacy is lower than that of the CNN-based model by 0.14% for the restau-

Table 7
Comparison average performance of methods for three datasets (%).

Aspect-term SA with Word2Vec embeddings									
Method	Tweet dataset			Restaurant dataset			MAMS dataset		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	71.68	71.47	71.57	70.51	70.29	70.39	58.24	57.97	58.10
FEM+BiLSTM+CRF	72.06	71.70	71.88	71.49	71.34	71.41	58.55	59.39	58.96
BANN	73.01	72.41	72.71	73.93	73.25	73.59	59.34	60.12	59.72
CNN-based	71.24	70.87	71.05	70.63	70.45	70.53	56.97	57.27	57.11
CANN	71.93	71.67	71.80	73.06	73.12	73.09	59.55	58.58	59.06
CANN-SG	73.07	73.31	73.19	74.91	74.06	74.48	59.65	60.85	60.24
CANN-SKG	72.12	72.90	72.50	74.76	74.67	74.71	59.95	60.61	60.28
CANN-CG	72.82	73.30	73.05	74.12	74.25	74.18	60.32	60.15	60.23
CANN-SSG	74.89	73.92	74.40	75.69	75.04	75.35	61.48	60.93	61.20
CANN-SCG	74.71	75.10	74.90	75.08	76.08	75.57	61.29	61.41	61.34
CANN-SKCG	75.28	74.17	74.72	76.08	75.16	75.61	60.86	61.26	61.06
CANN-SSCG	**76.99**	**77.99**	**77.48**	**78.30**	**78.54**	**78.42**	**61.80**	**62.25**	**62.02**
Aspect-term SA with GloVe embeddings									
Method	Tweet dataset			Restaurant dataset			MAMS dataset		
	Precision	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*	*Precision*	*Recall*	F_1*score*
BiLSTM+CRF	71.43	71.52	71.47	71.22	70.83	71.02	59.69	58.36	59.02
FEM+BiLSTM+CRF	72.43	71.43	71.92	71.41	71.57	71.49	58.17	59.49	58.82
BANN	73.84	73.66	73.75	74.41	74.39	74.40	59.73	59.17	59.45
CNN-based	71.00	70.20	70.59	70.77	70.19	70.47	56.96	57.12	57.03
CANN	71.21	72.82	72.00	72.76	72.97	72.86	59.76	59.56	59.66
CANN-SG	74.72	74.62	74.67	74.93	74.22	74.57	60.43	59.59	60.00
CANN-SKG	74.44	75.12	74.77	74.57	75.69	75.12	60.35	60.79	60.56
CANN-CG	74.53	74.87	74.70	74.68	75.56	75.11	60.55	60.36	60.44
CANN-SSG	76.38	75.31	75.84	76.62	75.43	76.02	61.97	62.12	62.03
CANN-SCG	77.40	77.21	77.30	77.39	77.75	77.55	61.74	61.38	61.54
CANN-SKCG	77.87	76.69	77.27	77.99	78.06	78.02	61.37	61.27	61.32
CANN-SSCG	**78.90**	**79.22**	**79.06**	**79.59**	**79.90**	**79.74**	**62.50**	**62.60**	**62.55**

rant dataset. However, for the aspect-term SA with GloVe embeddings, the BiLSTM+CRF model can improve the F_1*score* of the CNN-based by at least 0.55% and by up to 1.99% for the tweet dataset and the MAMS dataset, respectively. These results indicate that GloVe can represent words better than word2vec and LSTM can learn better context representations. Among the LSTM-based models, the BiLSTM+CRF method performs the worst. In our assessment, in this model, the roles of targets and contexts are not clearly different.

- The use of an attention mechanism in the BANN and CANN models results in greater improvement. Attention mechanisms help retain significant information in the context and create better vectors for aspect-level SA. For example, when using GloVe as word embeddings, the BANN can improve the F_1*score* of BiLSM+CRF by up to 2.24% and that of FEM+BiLSTM+CRF by up to 1.83% for the tweet dataset. It can also increase the F_1*score* of FEM+BiLSTM+CRF and BiLSM+CRF on the restaurant dataset by 2.91% and 3.38%, respectively; and similarly for the remaining datasets and for aspect-term SA with word2vec embeddings.
- The CNN model with additional attention mechanisms (called the CANN) performs competitively and outperforms the original CNN method. For example, in the case of aspect-term SA with word2vec embeddings, the CANN model can increase the performance of the CNN-based method by at least 0.75% for the tweet dataset and by up to 2.56% for the restaurant dataset. Furthermore, in the case of the aspect-term SA with GloVe embeddings, the CANN model can increase the performance of the CNN-based method by at least 1.41% for the tweet dataset and by up to 2.63% for the MAMS dataset. This proves that the pooling layer attending the attention mechanism can well obtain the information and represent more diverse and more complicated features than other methods.
- For ablated models, in most datasets, the CANN algorithm over graphs integrated from two subgraphs, such as CANN-SSG, CANN-SCG, and CANN-SKCG, performs better than models using the CANN algorithm over separate subgraphs, such as CANN-SG, CANN-SKG, and CANN-CG. For example, in the case of the aspect-term SA with word2vec embeddings, the CANN-SKCG method has increased the F_1*score* of the CANN-CG and the CANN-SKG by at least 0.78% and 0.83% for the same MAMS dataset and by up to 2.22% and 1.67% for the same tweet dataset, respectively. However, in the case of aspect-term SA with GloVe embeddings, the CANN-SKCG method has increased the F_1*score* of the CANN-CG and the CANN-SKG by at least 0.88% and 0.76% for the same MAMS dataset and by up to 2.91% and 2.9% for the same restaurant

dataset, respectively. This indicates that simultaneously considering the impact of both syntactic and sentiment knowledge, both syntactic and context-specific knowledge, or sentiment knowledge and context-specific knowledge usually achieves better performance than integrating only one.

- Our CANN-SSCG model outperforms the other methods on most datasets, which proves that the SSC graph can capture syntactic, semantic, and contextual information better than other methods and is beneficial for aspect-level SA tasks. Compared with other attention mechanism-based baselines, such as the CANN and BANN, the results show that the CANN-SSCG model outperforms all models, except for attention mechanism-based models. The CANN-SSCG model can improve the F_1*score* of the BANN model on three datasets (tweet, restaurant, and MAMS) by 4.77%, 4.83%, and 2.3%, respectively, on aspect-term SA with word2vec embeddings and by 5.31%, 5.34%, and 3.1%, respectively, for aspect-term SA with GloVe embeddings. In addition, the CANN-SSCG model can improve the F_1*score* of the CANN model on three datasets (tweet, restaurant, and MAMS) by 5.68%, 5.33%, and 2.96%, respectively, for aspect-term SA with word2vec embeddings and by 7.06%, 6.88%, and 2.89%, respectively, for aspect-term SA with GloVe embeddings. These results indi-

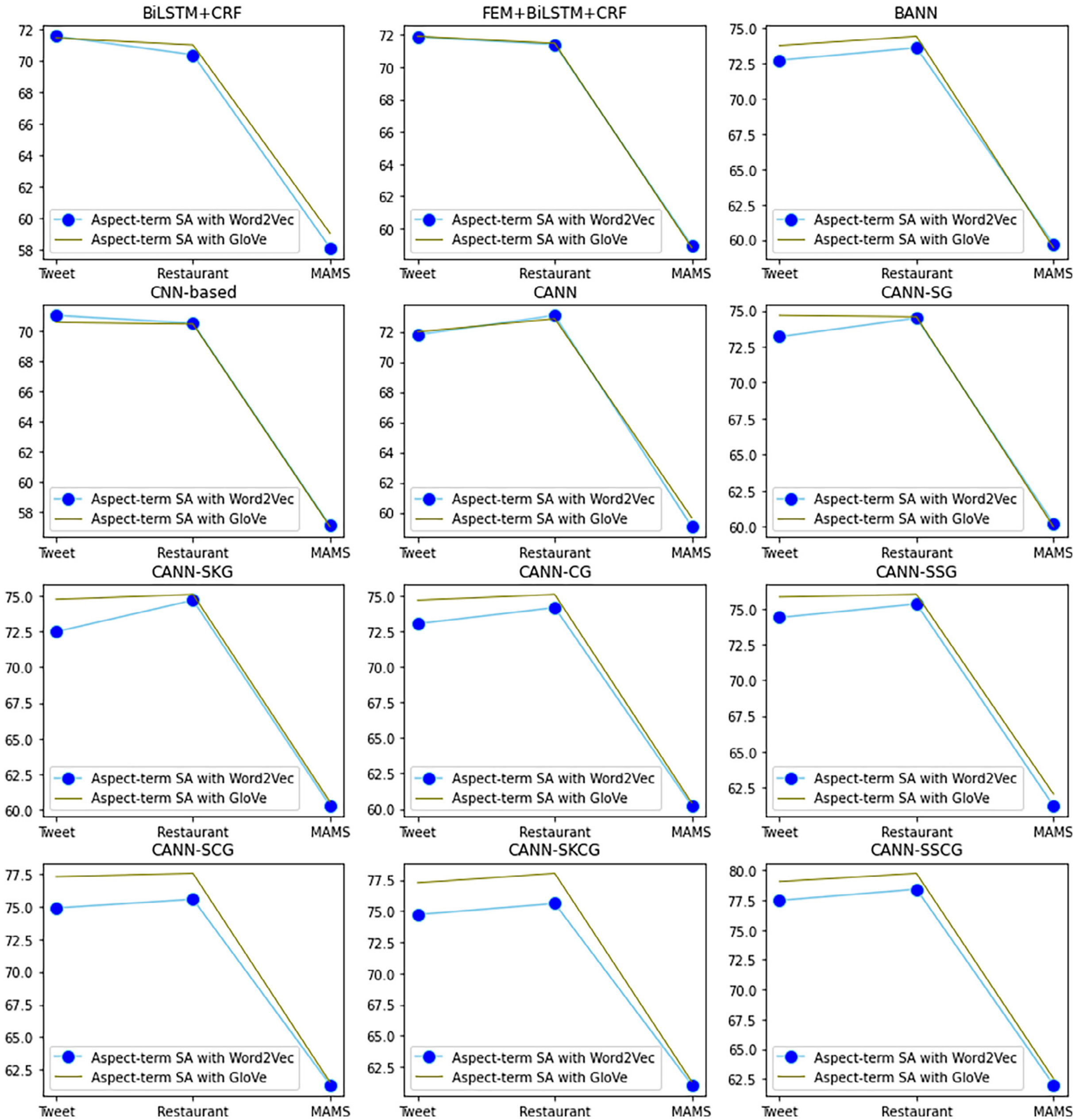


Fig. 6. The comparison toward F_1 score of methods for each separate word embeddings.

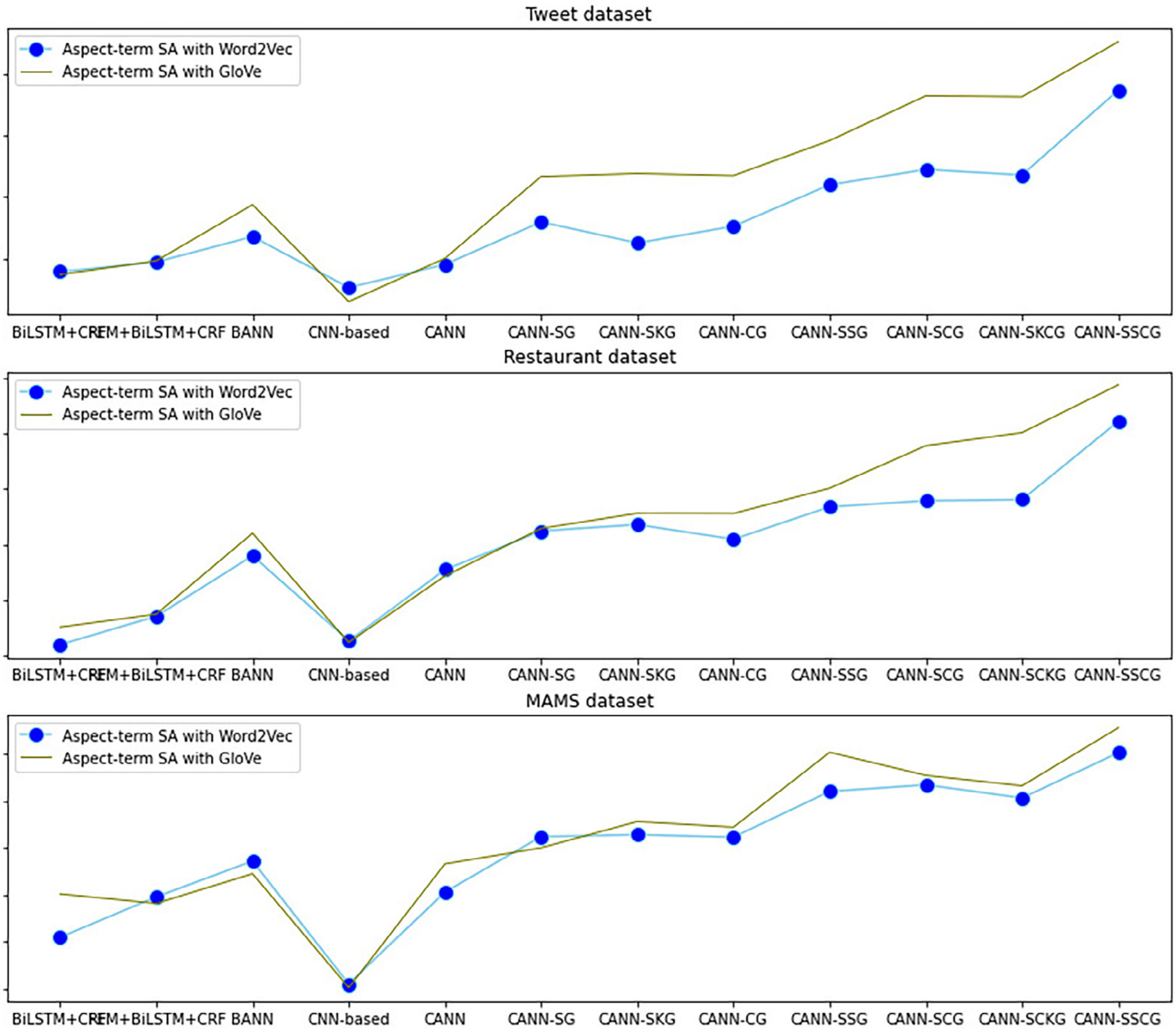


Fig. 7. The comparison toward F_1 score of methods on each separate dataset.

cate that using the BiLSTM model, the attention mechanism, and the GCN simultaneously can yield better performance for aspect-level SA than using each individually. As expected, the LSTM layer in BiLSTM is useful for representing information and dependency context in long texts such as sentences and documents. The convolutional layer and attention mechanism in CANNs are considered to be effective at collecting local and position features. The SSC graph can fully capture information regarding syntactic, sentiment knowledge, and context-specific aspects and their sentiments. Although the CANN-SSCG model achieves better results than the baselines, the $F_1 score$ does not increase significantly. This is caused by a mismatch in the POS tagging and the dependency parsing results not being sufficiently precise.

The abovementioned conclusions indicate that our CANN-SSCG model achieves promising performance for aspect-level SA compared to using each method individually. As expected, in BiLSTM, the input information can be accessed from the current state, and the output layer can obtain information from backwards and forwards hidden states, making BiLSTM especially helpful when the input is the context. In addition, the LSTM layers in BiLSTM are well suited to encoding information and long-range context dependency. Therefore, using BiLSTM to contextualize word representations can yield the best representations. Additionally, the SSC graph can fully capture and represent the features of syntactic, semantic, and contextual information regarding the position of aspects and their sentiment polarity. Finally, the convolutional layer and attention mechanism in the CANN can effectively extract local and position-invariant features to identify the positions of aspects and their sentiment polarity more accurately.

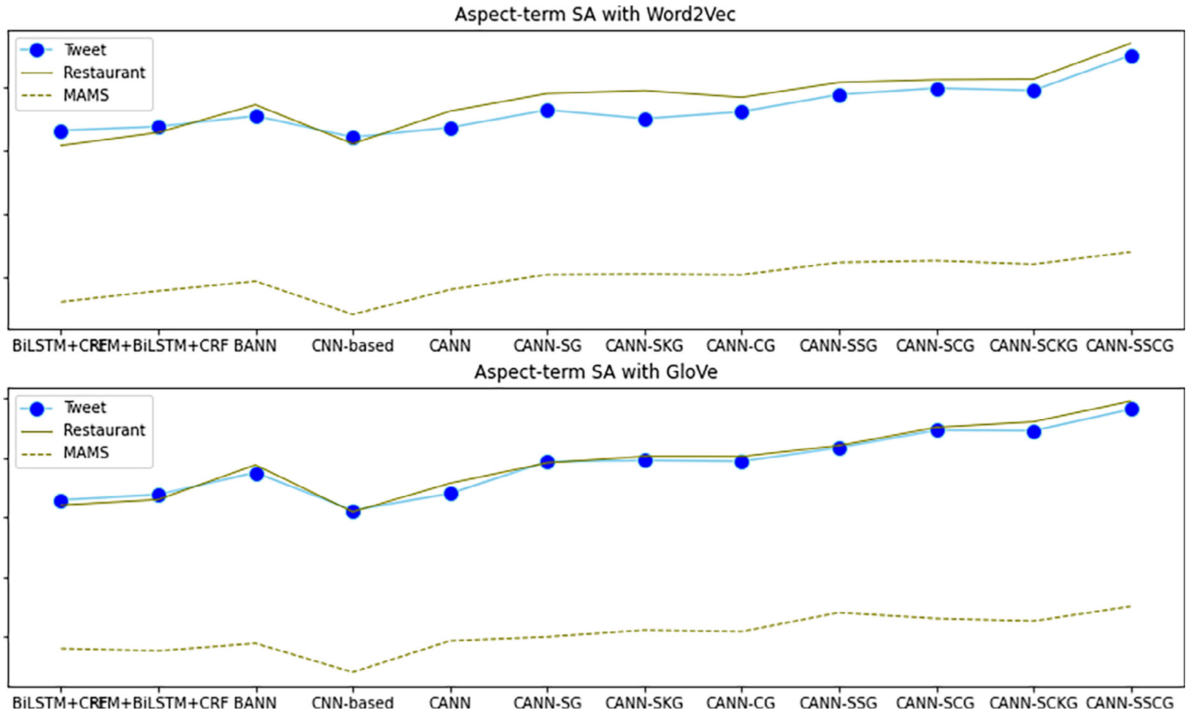


Fig. 8. The comparison toward F_1 score of separate methods on all three datasets.

Table 8
Some observations toward the performance of methods for all datasets (H: High performance; M: Medium performance; L: Low performance).

Aspect-term SA with Word2Vec embeddings									
Methods	Tweet			Restaurant			MAMS		
	H	M	L	H	M	L	H	M	L
BiLSTM+CRF	√				√				√
FEM+BiLSTM+CRF	√				√				√
BANN		√		√					√
CNN-based	√				√				√
CANN		√		√					√
CANN-SG		√		√					√
CANN-SKG		√		√					√
CANN-CG		√		√					√
CANN-SSG		√		√					√
CANN-SCG		√		√					√
CANN-SKCG		√		√					√
CANN-SSCG		√		√					√

Aspect-term SA with GloVe embeddings									
Methods	Tweet			Restaurant			MAMS		
	H	M	L	H	M	L	H	M	L
BiLSTM+CRF	√				√				√
FEM+BiLSTM+CRF	√				√				√
BANN		√		√					√
CNN-based	√				√				√
CANN		√		√					√
CANN-SG	√				√				√
CANN-SKG		√		√					√
CANN-CG		√		√					√
CANN-SSG		√		√					√
CANN-SCG		√		√					√
CANN-SKCG		√		√					√
CANN-SSCG		√		√					√

6. Conclusion and future works

The primary purpose was to develop a methodology to improve the performance of aspect-level SA via the CANN over an SSC graph structure (referred to as CANN-SSCG). In this study, the attractive advantage of the combination of deep learning algorithms and graph structures (e.g., the CANN and GCN)'that it can simultaneously learn multiple pieces of information when analysing aspect-level sentiment'was discovered and applied to real datasets. Three novel efficient GCN-based architectures'syntax-based, semantic-based, and context-based graphs'were proposed to learn syntactic information, knowledge regarding sentiment, and contextual importance for significant feature representation. A new GCN-like model, called the SSC graph, was then constructed to integrate the most important information learned from the previous three graphs for sentence representation. Finally, a CNN-based algorithm with an attention mechanism regarding the position of words indicating aspects, namely, the CANN, was developed to predict the position of aspects and their sentiment for aspect-level sentiment classification.

The aspect-level sentiment classification of the proposed CANN-SSCG was compared with that of popular methods on three real datasets. The experimental results show that the proposed model can improve the F_1score. The research advantages of this study are summarized as follows:

- Unlike other graph structures, the proposed SSC graph considers the dependence relationship between the word nodes, the co-occurrence information between the tokens, and the sentiment scores of these words. Additionally, it considers the impact of word co-occurrence and the relationships between sentence words and among sentences.
- The proposed method (CANN-SSCG) completely outperforms the baseline methods with an F_1score up to 7.75% on the aspect-level SA task.
- Owing to the importance of the three subgraphs and attention mechanism, the CANN-SSCG model achieves much better performance than using GCNs and CANNs individually.
- The proposed CANN-SSCG model is more effective than the ablated models. In practice, this model can easily and simultaneously use various types of information regarding aspects and sentiment. Therefore, the proposed model can achieve better performance.

In the future, we intend to use the CANN-SSCG model to support decision-making by considering contextual analysis. We also hope to use the CANN-SSCG model as an application prototype to solve the problem of fake news detection on a social network.

CRediT authorship contribution statement

Huyen Trang Phan: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Ngoc Thanh Nguyen:** Supervision, Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **Dosam Hwang:** Supervision, Methodology, Project administration, Funding acquisition.

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.

References

[1] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, K. Sima'an, Graph convolutional encoders for syntax-aware neural machine translation, 2017, arXiv preprint arXiv:1704.04675..
[2] K. Bijari, H. Zare, E. Kebriaei, H. Veisi, Leveraging deep graph-based text representation for sentiment polarity applications, Expert Systems with Applications 144 (2020) 113090.
[3] E. Cambria, J. Fu, F. Bisio, S. Poria, Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015.
[4] E. Cambria, Y. Li, F.Z. Xing, S. Poria, K. Kwok, Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 105–114.
[5] E. Cambria, S. Poria, D. Hazarika, K. Kwok, Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
[6] R. Collobert, Deep learning for efficient discriminative parsing, in: G. Gordon, D. Dunson, M. Dudík (Eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 2011, pp. 224–232. URL: http://proceedings.mlr.press/v15/collobert11a.html..
[7] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Advances in Neural Information Processing Systems 29 (2016) 3844–3852.
[8] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: The 52nd Annual Meeting of the Association for Computational Linguistics (ACL), ACL, 2014.
[9] G. Ganu, N. Elhadad, A. Marian, Beyond the stars: improving rating predictions using review text content, in: WebDB, Citeseer, 2009, pp. 1–6..
[10] A. García-Pablos, M. Cuadros, G. Rigau, W2vlda: almost unsupervised system for aspect based sentiment analysis, Expert Systems with Applications 91 (2018) 127–137.
[11] R. He, W.S. Lee, H.T. Ng, D. Dahlmeier, Effective attention modeling for aspect-level sentiment classification, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1121–1131.
[12] F. Hemmatian, M.K. Sohrabi, A survey on classification techniques for opinion mining and sentiment analysis, Artificial Intelligence Review 52 (2019) 1495–1545.
[13] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, 2015, arXiv preprint arXiv:1508.01991..
[14] Y. Jacob, Y. Winetraub, G. Raz, E. Ben-Simon, H. Okon-Singer, K. Rosenberg-Katz, T. Hendler, E. Ben-Jacob, Dependency network analysis (d ep na) reveals context related influence of brain network nodes, Scientific Reports 6 (2016) 1–19.
[15] Q. Jiang, L. Chen, R. Xu, X. Ao, M. Yang, A challenge dataset and effective models for aspect-based sentiment analysis, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 6281–6286.
[16] Y.N. Kenett, D.Y. Kenett, E. Ben-Jacob, M. Faust, Global and local features of semantic networks: Evidence from the hebrew mental lexicon, PloS One 6 (2011) e23912.
[17] Y. Kim, Convolutional neural networks for sentence classification, 2014, CoRR abs/1408.5882. URL: http://arxiv.org/abs/1408.5882, arXiv:1408.5882..
[18] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980..
[19] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907..
[20] A. Kumar, V.T. Narapareddy, V.A. Srikanth, L.B.M. Neti, A. Malapati, Aspect-based sentiment classification using interactive gated convolutional network, IEEE Access 8 (2020) 22445–22453.

[21] Y. LeCun, et al., Lenet-5, convolutional neural networks, 2015, URL: http://yann.lecun.com/exdb/lenet, 20, 14..
[22] H. Liu, T. Zhu, F. Shang, Y. Liu, D. Lv, Y. Shuyuan, Deep fuzzy graph convolutional networks for polsar imagery pixel-wise classification, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2020).
[23] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
[24] D. Marcheggiani, J. Bastings, I. Titov, Exploiting semantics in neural machine translation with graph convolutional network, in: 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2018, pp. 486–492.
[25] D. Marcheggiani, I. Titov, Encoding sentences with graph convolutional networks for semantic role labeling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1506–1515.
[26] H.T. Nguyen, L.M. Nguyen, Ilwaanet: An interactive lexicon-aware word-aspect attention network for aspect-level sentiment classification on social networking, Expert Systems with Applications 146 (2020) 113065.
[27] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, in: International Conference on Machine Learning, PMLR, 2016, pp. 2014–2023.
[28] S.O. Oppong, D. Asamoah, E.O. Oppong, D. Lamptey, Business decision support system based on sentiment analysis, International Journal of Information Engineering and Electronic Business 12 (2019) 36.
[29] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, Q. Yang, Large-scale hierarchical text classification with recursively regularized deep graph-cnn, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1063–1072.
[30] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[31] H.T. Phan, N.T. Nguyen, V.C. Tran, D. Hwang, A sentiment analysis method of objects by integrating sentiments from tweets, Journal of Intelligent & Fuzzy Systems 37 (2019) 7251–7263.
[32] H.T. Phan, N.T. Nguyen, V.C. Tran, D. Hwang, An approach for a decision-making support system based on measuring the user satisfaction level on twitter, Information Sciences 561 (2021) 243–273.
[33] H.T. Phan, V.C. Tran, N.T. Nguyen, D. Hwang, Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model, IEEE Access 8 (2020) 14630–14641.
[34] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, Semeval-2015 task 12: Aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 486–495..
[35] A. Pouran Ben Veyseh, N. Nouri, F. Dernoncourt, Q.H. Tran, D. Dou, T.H. Nguyen, Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation, Findings of the Association for Computational Linguistics: EMNLP 2020, 2020..
[36] W. Quan, Z. Chen, J. Gao, X.T. Hu, Comparative study of cnn and lstm based attention neural networks for aspect-level opinion mining, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 2141–2150.
[37] T. Rocktäschel, E. Grefenstette, K.M. Hermann, T. Kočiskỳ, P. Blunsom, Reasoning about entailment with neural attention, 2015, arXiv preprint arXiv:1509.06664.
[38] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 379–389.
[39] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1715–1725.
[40] K. Shuang, Q. Yang, J. Loo, R. Li, M. Gu, Feature distillation network for aspect-based sentiment analysis, Information Fusion (2020).
[41] Z. Toh, J. Su, Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 282–288..
[42] U. Von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (2007) 395–416.
[43] S. Wang, M. Huang, Z. Deng, Densely connected cnn with multi-scale feature attention for text classification, in: IJCAI, 2018, pp. 4468–4474..
[44] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based lstm for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.
[45] Z. Wu, Y. Li, J. Liao, D. Li, X. Li, S. Wang, Aspect-context interactive attention representation for aspect-level sentiment classification, IEEE Access 8 (2020) 29238–29248.
[46] Y. Xing, C. Xiao, Y. Wu, Z. Ding, A convolutional neural network for aspect sentiment classification, 2018, arXiv preprint arXiv:1807.01704..
[47] X. Yan, T. Ai, M. Yang, H. Yin, A graph convolutional neural network for classification of building patterns using spatial vector data, ISPRS Journal of Photogrammetry and Remote Sensing 150 (2019) 259–273.
[48] M. Yang, W. Tu, J. Wang, F. Xu, X. Chen, Attention based lstm for target dependent sentiment classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017.
[49] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 7370–7377.
[50] J. Zhou, J.X. Huang, Q.V. Hu, L. He, Sk-gcn: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification, Knowledge-Based Systems 205 (2020) 106292.