Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.048486

ARTICLE

Aspect-Level Sentiment Analysis Based on Deep Learning

Mengqi Zhang[1], Jiazhao Chai[2], Jianxiang Cao[3], Jialing Ji[3] and Tong Yi[4,*]

[1]Department of Communication, University of New South Wales, Sydney, 2052, Australia

[2]Higher Vocational Education Publishing Division, Higher Education Press, Beijing, 100029, China

[3]State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, 100020, China

[4]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin, 541004, China

*Corresponding Author: Tong Yi. Email: yitong@mailbox.gxnu.edu.cn

Received: 09 December 2023 Accepted: 22 January 2024 Published: 26 March 2024

ABSTRACT

In recent years, deep learning methods have developed rapidly and found application in many fields, including natural language processing. In the field of aspect-level sentiment analysis, deep learning methods can also greatly improve the performance of models. However, previous studies did not take into account the relationship between user feature extraction and contextual terms. To address this issue, we use data feature extraction and deep learning combined to develop an aspect-level sentiment analysis method. To be specific, we design user comment feature extraction (UCFE) to distill salient features from users' historical comments and transform them into representative user feature vectors. Then, the aspect-sentence graph convolutional neural network (ASGCN) is used to incorporate innovative techniques for calculating adjacency matrices; meanwhile, ASGCN emphasizes capturing nuanced semantics within relationships among aspect words and syntactic dependency types. Afterward, three embedding methods are devised to embed the user feature vector into the ASGCN model. The empirical validations verify the effectiveness of these models, consistently surpassing conventional benchmarks and reaffirming the indispensable role of deep learning in advancing sentiment analysis methodologies.

KEYWORDS

Aspect-level sentiment analysis; deep learning; graph convolutional neural network; user features; syntactic dependency tree

1 Introduction

Deep learning [1] revolutionizes machine learning by automating complex feature extraction. Deep learning methods have been widely used in various fields, e.g., human activity recognition [2], disease detection [3], and traffic flow prediction [4]. As deep neural networks advanced, improved attention mechanisms and diverse word vector representations enhanced the role of deep learning in sentiment analysis. Among the sentiment analysis tasks, aspect-level sentiment analysis is the most fine-grained sentiment analysis method. It takes the opinion target in the sentence as the research object and identifies its sentiment polarity (e.g., positive, negative, or neutral) [5]. Opinion targets can be divided into entity and aspect. The entity is a specific object that the sentence describes, such as a product, a service, or an event. An aspect is the attribute of an entity, such as the pixels of a mobile phone, the environment of a restaurant, etc.

In the early years, traditional aspect based sentiment analysis (ABSA) methods relied primarily on rule-based or feature-based machine learning models, which often required a lot of human intervention and were limited in their effectiveness when dealing with complex linguistic features and implicit expressions of emotions, mainly including dictionary-based approaches and traditional machine learning approaches. Dictionary-based approaches are mainly based on sentiment dictionaries and language rules to identify sentiment polarity [6–8]. However, with the development of the times, the number of new words on the Internet almost explodes, leading to the difficulty of including these new words. Nowadays, dictionary-based approaches are no longer used alone, but instead integrated with deep learning approaches [9]. Traditional machine learning approaches automatically learn rules from sentences, mainly including linear models, support vector machines, decision trees, etc. [10–12]. In the face of such problems, modern deep learning techniques, particularly those rooted in neural networks, offer promising avenues for handling intricate sentiment analysis tasks. Deep learning approaches do not need to manually design features and can maintain rich semantic information through the automatically trained vectors, which greatly improves the accuracy of aspect-level sentiment classification. Deep learning approaches have better performance than the dictionary-based approaches and the traditional machine learning approaches [13–15]. Therefore, this paper uses the deep learning-based approach to conduct research.

However, the above deep learning research on aspect-level sentiment analysis only focuses on texts and does not involve users who post comments. Since the subjectivity of the users can have a profound impact on the sentiment tendency of each comment, we are considering making some improvements to the deep learning methods to solve this problem. In addition, the existing deep learning methods only focus on the relationship between words but ignore the deep semantic connection between aspect words. This paper introduces the User-aspect-sentence Graph Convolution Neural Network model (U-ASGCN) as a novel approach, leveraging the power of deep learning, which combines text features and user features to predict the sentiment polarity of aspect words. The main contributions of this paper are threefold:

1. A pioneering deep learning model, user comment feature extraction (UCFE), is presented to enhance the aspect-level sentiment analysis. The model extracts the common features from all historical comments of users and transforms them into user feature vectors. The experimental results show that the UCFE deep learning method used in this paper can extract users' subjective features more accurately and effectively.

2. Aspect-sentence Graph Convolutional Network (ASGCN) deep learning method is introduced to further obtain context information, and the powerful capability of deep learning networks is used to obtain the relationship between aspect-oriented words. The model focuses on the differences between syntactic dependency types and introduces the relationship between aspect words in the modeling process, which can capture richer semantics in sentences. The experimental results show that the ASGCN model is superior to the baseline model.

3. To combine UCFE and ASGCN, we devise three feature embedding methods, where an aspect-level sentiment analysis model, namely U-ASGCN is derived. U-ASGCN can further improve the model performance compared with ASGCN.

The rest of this paper is organized as follows. We review and analyze the related work in Section 2 and our aspect-level sentiment analysis scheme is presented in Section 3. To validate the proposed scheme, UCFE, ASGCN, and U-ASGCN are evaluated in Section 4, respectively. Finally, we conclude this paper and provide the future research thinking in Section 5.

2 Related Works

2.1 Research of User Features

The most common way to use user information is to introduce a vector of user features into the model. Zhong et al. proposed a lightweight small sample learning method named UserAdapter based on the Transformer model [16]. Fatemehsadat et al. proposed the UserIdentifier model based on UserAdapter, which connects the user ID with the text to generate the augmented matrix and associates the user embedding parameters with the parameters of the Transformer model to obtain the user feature vector [17].

At present, some researchers use user features to study the sentiment tendency problem. For example, Christopher et al. used the geographical location of users to analyze the sentiment tendencies and dynamics between different cities and proposed a sentiment normalization method to narrow the differences in absolute sentiment standards between different cities [18]. Amin et al. proposed a neural network classifier based on users' psychological behavior to predict the existence of extreme users [19]. Their studies took the number of friends, followers, and comments of users as user features, calculated the sentiment score of each user, and used a multi-layer perceptron model to label the users' types. Another study by Amin et al. put forward a data-driven model to predict the changing trend of user sentiment in a period [20].

However, there is no research on user features in the aspect-level sentiment analysis. In this paper, we propose a UCFE model to extract user feature vectors and three methods to embed user feature vectors into the sentiment analysis model to play an important role in user features in sentiment analysis.

2.2 Aspect-Level Sentiment Analysis Based on GCN

The topological structure of Graph Convolution Neural Network (GCN) can model the relationship between words and dig potential information in sentences, which makes GCN widely used in the research of classification, dialogue, recommendation, and other fields.

In the problems of aspect-level sentiment analysis, GCNs are usually combined with syntactic dependency trees to capture the sentiment dependency relationships in sentences. Chen et al. found that convolutional neural networks would incorrectly identify grammatically irrelevant contexts as affective cues associated with aspect words [21]. To avoid such errors, a GCN model based on a syntactic dependency tree is proposed to capture the sentiment tendency of aspect words. Amir et al. proposed a graph-based deep learning model to obtain the importance rating of words from the dependency tree of sentences [22]. This model generates gate vectors from presentation vectors of aspect words and then builds the hidden vectors of the graph model focused on aspect words. Zhao et al. proposed a new aspect-level sentiment classification model based on graph convolutional networks to capture the sentiment dependence between multiple aspect-level words [23]. At the same time, using the GCN model on the improved attention mechanism, they proposed the first method for modeling semantic associations between aspect words. Wang et al. innovatively proposed a method for encoding syntactic information based on the GCN model [24]. The Relation Graph Attention Network (R-GAN) was proposed to encode the specific dependency tree. Tian et al. argued that the dependency types between words are also an important factor in the formation of dependency relationships between words, so they proposed a Type Graph Convolutional Neural Network (T-GCN) model to explicitly utilize the dependency relationships between aspects and use attention scores to distinguish different edges in the GCN [25].

Other researchers improved the structure of GCN and proposed some model frameworks more suitable for aspect-level sentiment analysis. Li et al. proposed a Dual Graph Convolutional Network model (DualGCN) considering syntactic structure complementarity and semantic structure dependency [26]. Meng et al. proposed a Weighted Graph Convolutional Neural Network (WGCN) to obtain rich syntactic information by combining features [27]. Their model takes the sentence representation vector generated by Bidirectional Encoder Representations from Transformers (BERT) as input and uses an alignment to solve the problem of vector inconsistencies between WGCN and BERT. Liang et al. built a GCN model named SenticGCN by integrating sentiment knowledge of SenticNet to enhance the dependency graph of sentences [28].

However, the above studies ignored the association between aspect words. Therefore, this paper captures the relationships among aspect words to improve performance.

3 U-ASGCN Model

We present our method in this section, and the description of the main symbols in this paper is shown in Table 1. To begin with, the problem definition and technical overview are provided. Then, the modules in our method are elaborated in the following subsections.

Table 1: Notations in U-ASGCN

Notations	Definition
$\{w_1, w_2, ..., w_k\}$	A sentence with k words
w_i	A word
$\{a_1^n, a_2^n, ..., a_m^n\}$	The aspect words of the n-th sentence
u_q	The ID of the q-th user
y_m^n	The sentiment polarity of the m-th aspect word in the n-th sentence

3.1 Task Definition

Let the model input the text set $S = (s_1, \ldots, s_i, \ldots, s_n)$ and the user set $U = (u_1, u_2, \ldots, u_q)$. From s_1 to s_i come from the same comment document. $s_n = (w_1, w_2, \ldots, w_k)$ represents the sentence, w_k represents the word of a sentence, and k is the number of words in the sentence, i.e., the length of the sentence. $a^n = \{a_1^n, a_2^n, \ldots, a_m^n\}$ represents the collection of aspect words of the n-th sentence, and m is the number of aspect words in the sentence. In the user set U, u_q represents the ID of the q-th user, $q \leq n$. In the experiment, the input of this paper is a set composed of several triples (u_q, a^n, s_n).

The output of this text is $Y = (y_1, y_2, \ldots, y_n)$, where $y_n = (y_1^n, y_2^n, \ldots, y_m^n)$, and y_m^n represents the sentiment polarity of the m-th aspect word in the n-th sentence. $y_m^n \in \{-1, 0, 1\}$, where -1 denotes negative sentiment, 0 denotes ambivalent or neutral sentiment, and 1 denotes positive sentiment.

3.2 The Overview of the U-ASGCN Model

The technical overview of our method is shown in Fig. 1. U-ASGCN mainly consists of two modules: the ASGCN and UCFE models. The UCFE model is used to extract the common features of the historical comments and represent the features as u_q. The ASGCN model is used to capture the semantic features of the sentences and train the representation vector of the comments. The user feature vector derived by UCFE is embedded into the ASGCN model to obtain the text output vector H^{output}, which integrates the user features. Finally, it is used for sentiment classification.

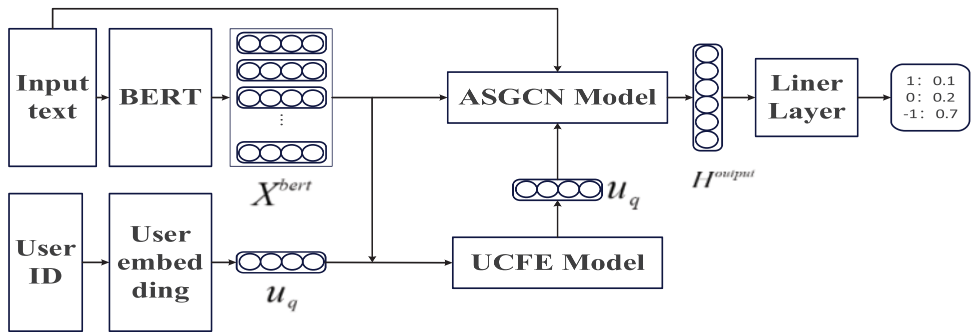

Figure 1: The model architecture of U-ASGCN

3.3 UCFE Model

The structure of UCFE is shown in Fig. 2.

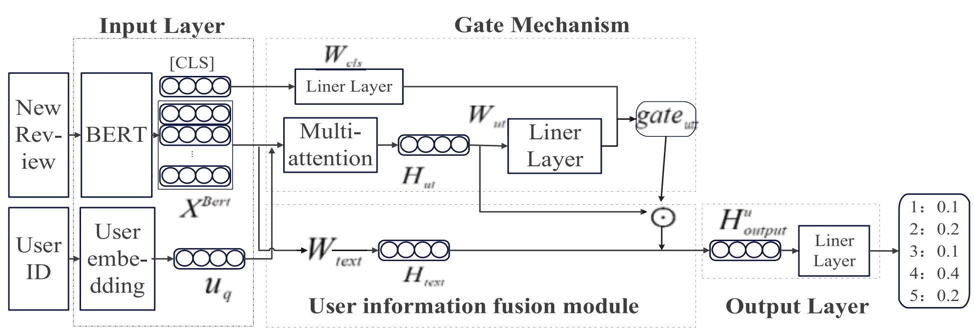

Figure 2: The model architecture of UCFE

To verify the performance of the UCFE model, this paper applies the UCFE model to the document-level sentiment classification scenario. Therefore, an additional user information fusion module and output layer are designed for the UCFE model to obtain the text output vector of UCFE for each comment.

3.3.1 Input Layer

The input layer uses the BERT model to train word embedding vectors, transforming the users' comment text into vectors, as shown in Eqs. (1) and (2):

$$\mathrm{X}^{\mathrm{Bert}} = \mathrm{Bert}\,(s_n) \tag{1}$$

$$\mathrm{X}^{\mathrm{Bert}} = (x_{cls}, x_1, x_2, \cdots, x_k) \tag{2}$$

$X^{Bert} \in R^{(k+1)\times embedding\_\dim}$, *embedding_*dim is the dimension of a word embed vector and x_{cls} is a vector representation of [CLS]. The parameter x_{cls} is initialized to a zero vector, and as the training progresses, it can be regarded as the vector representation of the entire comment. The user embedding layer transforms the user ID into a user vector representation u_q, $u_q \in R^{1\times embedding\_\dim}$.

3.3.2 Gate Mechanism

The gate mechanism uses the multi-head attention mechanism to add the user feature information to the text representation of the current comment to generate the text representation of a specific user H_{ut}. The calculated vector H_{ut} integrates the features of the user comments including the current comment. The process of calculating is shown in Eqs. (3) and (4):

$$H_{ut} = MultiHead\left(u_q, X^{Bert}, X^{Bert}\right) = Concat\left(head_1, \cdots, head_i\right) W^O \frac{\Delta y}{\Delta x} \tag{3}$$

$$head_i = soft\max\left(\frac{\left(u_q W^Q\right)\left(X^{Bert} W^K\right)^T}{\sqrt{d_k}}\right)\left(X^{Bert} W^V\right) \tag{4}$$

where $W^Q \in R^{embedding\_\dim\times d_k}$, $W^K \in R^{emdedding\_\dim\times d_k}$, $W^V \in R^{embedding\_\dim\times d_k}$ and $W^O \in R^{\in R^{emdedding\_\dim\times emdedding\_\dim}}$. Note that, $d_k = embedding\_\dim/head\_num$, $head\_num$ are the number of heads of the multi-headed attention mechanism, and $H_{ut} \in R^{1\times embedding\_\dim}$.

Inspired by the gate mechanism in the LSTM model, this paper sets up the user information gate. The calculation of the user information gate is shown in Eq. (5).

$$gate_{ut} = sigmoid\left(\left(W_{cls} {x_{cls}}^T + W_{ut} {H_{ut}}^T + b\right)^T\right) \tag{5}$$

where $x_{cls} \in R^{1*embedding\_\dim}$, $W_{cls} \in R^{embedding\_\dim*embedding\_\dim}$, $b \in R^{embedding\_\dim*1}$, and $gate_{ut} \in R^{1*embedding\_\dim}$. The user information gate is used to control how much user feature information is added to the text features.

3.3.3 User Information Fusion Module and Output Layer

The user information fusion module uses the user information gate to fuse the user information with the text vector of the current comment and then generates an output vector of the text. In the user information fusion module, X^{Bert} needs to be reduced and linear, and the word vector is integrated into the text vector. The dimension reduction operation is shown in Eq. (6).

$$x_i = \sum_{j\in[1,embedding\_\dim]} x_{i,j}, i \in [0, k+1] \tag{6}$$

After the dimension reduction, $X^{Bert} \in R^{1*(k+1)}$. The linear calculation of X^{Bert} after the dimension reduction is carried out as shown in Eq. (7).

$$H_{text} = \left(W_{text}\left(X^{Bert}\right)^T + b_{text}\right)^T \tag{7}$$

where $W_{text} \in R^{embedding\_\dim*(k+1)}$, $b_{text} \in R^{embedding\_\dim*1}$, and $H_{text} \in R^{1*embedding\_\dim}$. Eq. (8) fuses user information and text information into an output vector H^u_{output}.

$$H^u_{output} = H_{text} + gate_{ut} \odot H_{ut} \tag{8}$$

where $\odot$ represents multiplication of matrix elements. $H^u_{output} \in R^{1*embedding\_\dim}$. H^u_{output} is for prediction as shown in Eq. (9).

$$y = \text{argmax}\left(soft\max\left(W^u_{output} {H^u_{output}}^T + b^u_{output}\right)\right) \tag{9}$$

where $W^{u}_{output} \in R^{embedding\_\mathrm{dim} \times embedding\_\mathrm{dim}}$, $W^{u}_{output} \in R^{embedding\_1}$, and $y \in \{-1, 0, 1\}$.

3.3.4 Update of User Feature Vectors

The user feature vectors should include the common features of users' historical comment texts, so the user feature vector needs to be updated after each prediction. The update method is shown in Eq. (10).

$$u_q^{new} = u_q + \alpha_u H_{ut} \tag{10}$$

where α_u is the hyperparameter, which determines how much user history information is left, and its value is constantly updated during training.

3.4 ASGCN Model

The ASGCN model is divided into four parts: the input layer, the AspectsGCN sub-model, the SentenceGCN sub-model and the output layer. The model architecture is shown in Fig. 3.

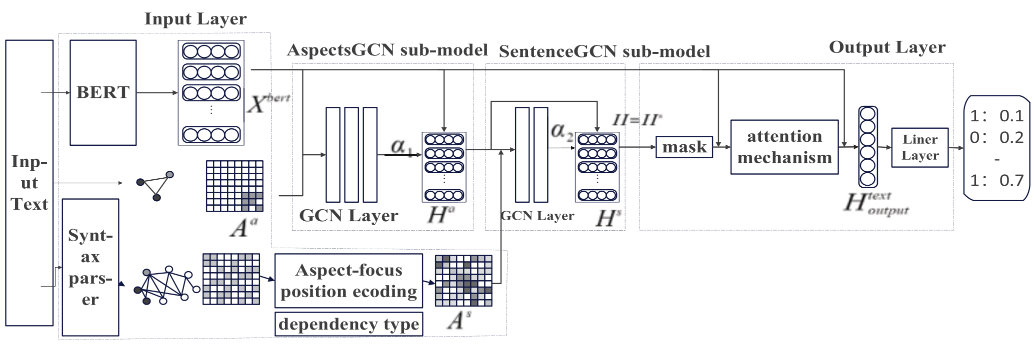

Figure 3: The model architecture of ASGCN

3.4.1 Input Layer

The input layer uses the BERT model to learn the embedding vector of words and builds a syntactic dependency graph between words and a relationship graph between aspect words for each sentence. The graph structure is stored in the adjacency matrix A^s and A^a, respectively.

First, build an adjacency matrix A^{dep} for each sentence based on the results of the syntactic parser, as shown in Eq. (11).

$$A_{i,j}^{dep} = \begin{cases} 1, i = jOR\omega_i \propto \omega_j \\ 0, others \end{cases} \tag{11}$$

where i and j correspond to the word's position in the sentence, respectively. $\omega_i \propto \omega_j$ means there is a dependency relationship between ω_i and ω_j. However, it is too absolute to use only 0,1 to represent the relationship between words. To pay attention to the position relationship between words, this paper adds position coding to the adjacency matrix. The calculation of position coding is shown in Eq. (12).

$$A_{i,j}^{position} = \begin{cases} 1, \omega_i, \omega_j \in a_m^n \\ \dfrac{1}{\frac{1}{|j-i|+1}}, \omega_i \in a_m^n \; AND \; \omega_j \notin a_m^n \\ 0, \omega_i \notin a_m^n \; AND \; \omega_j \in a_m^n \end{cases} \tag{12}$$

The meaning of Eq. (12) is that if and only if ω_i and ω_j are the same word or belong to the same aspect word, $A_{i,j}^{position} = 1$. When one of ω_i and ω_j is an aspect word, the position encoding is inversely proportional to the distance. When neither ω_i and ω_j is an aspect word, the position code is 0. That is, the above calculation of position coding is only for aspect words and does not consider the position relationship between two non-aspect words. The adjacency matrix A'^s with the position code is shown in Eq. (13).

$$A'^s = A^{dep} + A^{position} \tag{13}$$

Since aspect words in a sentence may share the same sentiment tendency, the relationship between one aspect word and other words (especially sentiment words) can provide a reference for the classification of the sentiment polarity of another aspect word. Therefore, on the premise of taking one of the aspect words as the center (central aspect), to pay more attention to other aspect words, this paper provides the supplementary information, as shown in Eq. (14).

$$\tilde{A}^s_{center} = \tilde{A}^s_{center} + \frac{1}{m-1}\sum_{i=0}^{m-1}\left(\alpha \tilde{A}^s_{other_i}\right), m > 1 \tag{14}$$

$$\alpha = \frac{1}{|position_{other_i} - position_{center}| + 1} \tag{15}$$

where $A'^s{}_{center}$ denotes the adjacency matrix of the central aspect word, $A'^s{}_{other_i}$ denotes the adjacency matrix of the i-th aspect word in the sentence except the central aspect word, and m denotes the total number of aspect words in the sentence. $position_{other_i}$ and $position_{center}$ represent the positions of other aspect words and central aspect words in the sentence, respectively. The meaning of Eq. (14) is that the amount of supplementary information is inversely proportional to the number of aspect words. The more aspect words, the less information each aspect word provides. And the supplementary information is inversely proportional to the distance between the other aspect words and the central aspect word. The further away from the central aspect words, the less supplementary information. During the experiment, aspect words in the sentence are successively taken as central aspect words. To simplify the expression, $A'^s{}_{center}$ will be represented as A''^s.

Finally, considering that the dependency types will also affect the classification results, this paper converts the dependency types into vectors and constructs the dependency type matrix $A^{deptype}$ according to the dependency type vectors, as shown in Eq. (16).

$$A_{i,j}^{deptype} = \begin{cases} \dfrac{\exp\left(\sum_{z=0}^{\dim} v_{i,j,z}\right)}{\sum_{i,j=0}^{k}\exp\left(\sum_{z=0}^{\dim} v_{i,j,z}\right)}, \omega_i \propto \omega_j \\ 0, others \end{cases} \tag{16}$$

where $v_{i,j,z} \in V_{i,j}$ represents the vector of dependency types between ω_i and ω_j, dim represents the dimension of dependency type vector, i.e., $z = 1, 2, 3, \cdots, \dim$. The dependency type vector is randomly initialized at the beginning and then updated during training.

To sum up, the adjacency matrix A^s is calculated as Eq. (17).

$$A^s = \tilde{A}^s A^{deptype} \tag{17}$$

The adjacency matrix A^a studies the relationship between aspect words without considering non-aspect words. In this paper, it is assumed that in the same sentence, every two aspect words are related to each other. This paper constructs the full connection graph between aspect words. The matrix A'^a of the fully connected graph is shown in Eq. (18).

$$A'^a_{i,j} = \begin{cases} 1, \omega_i \in a^n_m AND \omega_j \in a^n_{m'} \\ 0, \omega_i \notin a OR \omega_j \notin a^n \end{cases} \tag{18}$$

where a^n_m denotes the m-th aspect word in the sentence, $a^n_{m'}$ denotes the m'-th aspect word in the sentence. When $a^n_m = a^n_{m'}$, ω_i and ω_j belong to the same aspect word.

The distance between aspect words is also an important factor, and studies have found that the aspect words tend to have the same sentiment polarity as their left aspect words. Therefore, the position-coding of aspect words is also introduced in A'^a, as shown in Eq. (19).

$$A^a_{i,j} = \begin{cases} \alpha \dfrac{1}{|p+j-i|+1}, \omega_i \in a^n_m, \omega_j \in a^n_{m'} AND\, i \leq j \\ (1-\alpha) \dfrac{1}{|p+j-i|+1}, \omega_i \in a^n_m, \omega_j \in a^n_{m'} AND\, i > j \\ 0, \omega_i \notin a^n OR \omega_j \notin a^n \end{cases} \tag{19}$$

where p denotes the position of the aspect word ω_j in the multi-word aspect word $a^n_{m'}$, α denotes the weight.

3.4.2 Aspects GCN

Aspects GCN uses the GCN layer to capture the relationships between aspect words, as shown in Eq. (20).

$$G^{l,a} = \mathrm{Re}\,Lu\left(A^a G^{l-1,a} W^l + b^l\right) \tag{20}$$

where $G^{l,a}$ represents the output of Layer l, $G^{\mathrm{O},a} = \mathrm{X}^{\mathrm{Bert}}$. The output of the last layer is denoted as $G^{l,a}$, $\mathrm{G}^{\mathrm{L},a} \in R^{(k+1)\times embedding\_\mathrm{dim}}$. $G^{l,a}$ captures the relationship between aspect words and then adds the relation between aspect words to the vector $\mathrm{X}^{\mathrm{Bert}}$, as shown in Eq. (21).

$$H^a = X^{Bert} + \alpha_1 G^{L,a} \tag{21}$$

where α_1 is the hyperparameter, which determines how much aspect-word information is added to X^{Bert}. H^α represents the text embedding vector that contains the relationships between aspect words.

3.4.3 Sentence GCN

Sentence GCN uses the GCN layer to capture syntactic dependencies between words, as shown in Eq. (22).

$$G^{l,s} = \mathrm{Re}\,Lu\left(A^s G^{l-1,s} W^l + b^l\right) \tag{22}$$

where $G^{l,s}$ represents the output of SentenceGCN at the layer 1, $G^{\mathrm{O},s} = H^a$. The output of the last layer is $G^{L,s}$. The multi-layer GCN loses information between aspects while paying attention to syntactic dependencies. Therefore, this paper sets up an information compensation mechanism (ICM) as shown in Eq. (23).

$$H^s = G^{L,s} + \alpha_2 G^{L,a} \tag{23}$$

where α_2 is the information compensation hyperparameter, which determines how much aspect word information is added to $G^{L,s}$. H^s represents the text embedding vector that contains the syntactic dependencies between words.

3.4.4 Masking and Attention Mechanism

To use more accurate semantic information, this paper uses the masking and attention mechanisms in the output layer. First, since the final prediction is for aspect words, to eliminate the interference of non-aspect words, this paper uses the masking mechanism to set the non-aspect words as 0, as shown in Eq. (24).

$$H^{mask} = mask\,(H) = \left(0, \cdots 0, h_\tau, \cdots, h_{\tau+p-1}, 0, \cdots, 0\right). \tag{24}$$

where h_τ represents the τ-th word in the sentence, and also the first word of the multi-word aspect. p is the length of the multi-word aspect.

This paper uses the attention mechanism to set the attention weight for each word, as shown in Eqs. (25) and (26).

$$\beta_t = \sum_{i=1}^{k} x_t^{Bert} h_i^{mask} = \sum_{i=\tau}^{\tau+p-1} x_t^{Bert} h_i^{mask} \tag{25}$$

$$\alpha_t^{attention} = \frac{\exp\,(\beta_t)}{\sum_{i=1}^{k} \exp\,(\beta_i)} \tag{26}$$

where x_t^{Bert} represents the word embedding vector of the t-th word, and $\alpha_t^{attention}$ represents the attention score of the t-th word.

Sentiment predictions are shown in Eqs. (27) and (28).

$$H_{output}^{text} = \sum_{t=1}^{k} \alpha_t^{attention} x_t^{Bert} \tag{27}$$

$$y = \text{argmax}\left(soft\max\left(W^{output} H^{output} + b^{output}\right)\right) \tag{28}$$

3.5 User Feature Embedding Method

This paper proposes three methods to embed user feature vectors into the ASGCN model.

Method 1: Embedding the features before the AspectsGCN sub-model of the ASGCN model. The idea is to embed user features before capturing the semantic information of the text to use user information through the entire model deeply. Method 1 is shown in Fig. 4.

Since the dimensions of user feature vectors u_i and X^{Bert} are inconsistent, dimension expansion is required first, as shown in Eq. (29).

$$U_i \leftarrow \begin{pmatrix} u_{i,1} & \cdots & u_{i,1} \\ \vdots & \ddots & \vdots \\ u_{i,embedding\_\dim} & \cdots & u_{i,embedding\_\dim} \end{pmatrix} \tag{29}$$

After that, embedding the user feature vector is shown in Eq. (30).

$$H^{user\_text} = X^{Bert} \cdot U_i \tag{30}$$

"·" represents matrix multiplication, $H^{user\_text} \in R^{K \times embedding\_\mathrm{dim}}$. The reason for matrix multiplication is that the row of X^{Bert} represents the word vector, and the column of U_i represents the user features. After matrix multiplication, each word vector has user features.

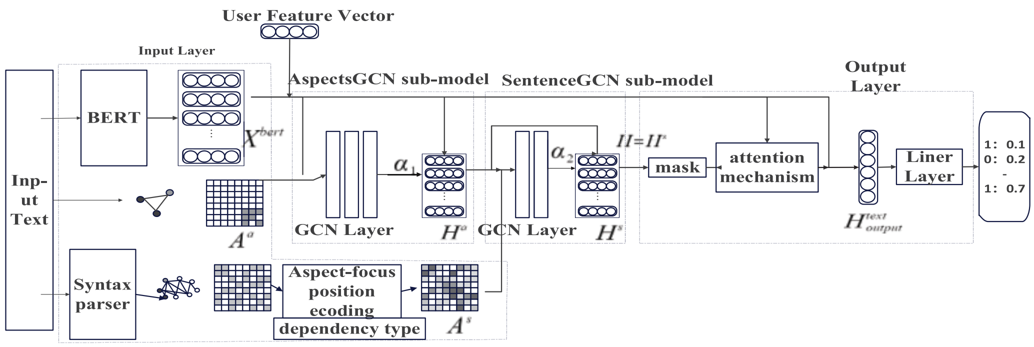

Figure 4: Embed before the AspectsGCN submodel

Method 2: Embedding the features after the SentenceGCN sub-model. The idea of this method is to embed user features in the prediction phase, preserving the complete text information and user features possible. Method 2 is shown in Fig. 5.

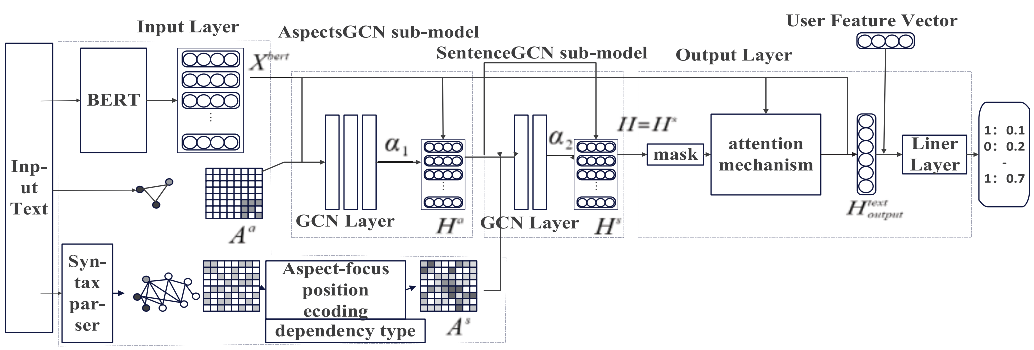

Figure 5: Embed after the SentenceGCN submodel

The embedding of user feature vectors is shown in Eq. (31).

$$H^{user\_text} = H^{output} + \alpha_3 u_i \tag{31}$$

where α_3 is the hyperparameter, which determines how much user information is embedded into the text output vector H^{output}. It is worth noting that in this method, the user feature vector does not need to be dimensionally extended. In this method, α_3 updates as the model training process to adaptively decide how many user features to embed based on the learning sample.

The third embedding method named the non-adaptive adjustment method that is similar to the second method. The difference is parameter α_3 in Eq. (31) is set to be a fixed value in the non-adaptive adjustment method.

4 Experiments Results and Analysis

We conduct the experiments to verify the practice of our method. Firstly, the performances of UCFE and ASGCN are evaluated independently. Then, we combine these two methods and analyze the effectiveness of U-ASGCN. We repeat every experiment three times independently, and only the average one is listed in the following experimental findings. Our methods are implemented by pycharm 2021 in Windows 10.

4.1 Validity Experiment of UCFE Model

4.1.1 Experiment Setup

To accurately reflect the commonalities of historical comments, this paper trains user feature vectors based on document-level sentiment classification tasks. Yelp Open Dataset[1] is the popular experiment dataset in sentiment classification. We use the comment data for 2013 and 2014 (Yelp-2013 and Yelp-2014) to evaluate our method. To make the dataset more consistent with the experimental requirements, only the user ID, comment text, and score are taken in this paper.

4.1.2 Baseline Models

We use four representative sentiment classification schemes as the baseline models that are grounded on combination vectors, Bidirectional Long Short-Term Memory (BiLSTM), and BERT, respectively. The details of the baseline models are as follows:

User-Word Representation Learning (UWRL) [29]: This model considers the influence of user features on text content. This model includes two parts: the user word combination vector model (UWCVM) and the document combination vector model (DCVM). Specifically, the UWCVM model modifies the meaning of words based on specific users, and DCVM takes the modified word vector as input to generate a text vector for prediction.

Hierarchical User Attention and Product Attention neural network (HUAPA) [30]: A document-level sentiment classification model based on the BiLSTM model. This model acquires the features of the user and the product, respectively. Specifically, the important features of the user or product are obtained through the attention mechanism, and then the user vector and product vector are fused respectively to model the text content of the document. Finally, this model generates two representation vectors of the document and connects the representation vectors to generate the final representation. In addition, a combination strategy is added to the model to enhance the text representation related to sentiment.

Chunk-wiseImportance Matrix (CHIM) [31]: An improved text classification model based on the BiLSTM model. This model transforms, mutates, and computes user and product information, and represents it as a block-weight matrix. The experimental results show that the method is significantly superior to the weighted method.

Review Rating Prediction with User and Product Memory (RRP-UPM) [32]: A document-level sentiment classification model based on the BERT model. This model trains feature vectors for user and product information, respectively. Specifically, the model trains feature vectors for all comments of the same user and all comments of the same product by explicit storage. In terms of model structure, RRR-UPM abandons hierarchy and directly correlates words in text.

4.1.3 Experimental Results and Analysis

From Table 2, it follows that UCFE's prediction is better than other baseline models. Longitudinally, the UCFE and RRP-UPM outperformed the HUAPA and CHIM. It is speculated that text

[1] https://www.yelp.com/dataset

features are more important than user features in document-level sentiment classification tasks. The text vectors obtained by the BERT model contain richer semantics. Therefore, the BERT-based model performs better than the BiLSTM-based model. Horizontally, all models perform better on the Yelp-2014 dataset than on the Yelp-2013 dataset. The accuracy of UCFE on the two datasets is only 0.3 percentage points different. This result shows that UCFE is robust.

Table 2: Experimental results of UCFE compared with the baseline model

Model	Yelp-2013		Yelp-2014	
	Acc	RMSE	Acc	RMSE
UWRL	61.8	96.2	——	——
HUAPA	68.3	62.8	68.6	62.6
CHIM	67.8	64.1	69.2	62.2
RRP-UPM	69.0	62.9	69.7	62.1
UCFE	70.2	58.3	70.5	57.9

4.2 Validity Experiment of ASGCN Model

4.2.1 Experiment Setup

To verify the practice of our method, four real-world and public datasets are used as follows: SemEval 2014's Restaurant comment dataset (Restaurant14)[2], SemEval 2014's Laptop comment dataset (Laptop14)[3], SemEval 2015's Restaurant comment dataset (Restaurant 15)[4], and SemEval 2016's Restaurant comment dataset (Restaurant 16)[5]. The word embedding vector in this paper is derived from the BERT model, and the dimension of the word embedding vector is set to 768 dimensions.

4.2.2 Baseline Model

To evaluate the performance of ASGCN, we use several advanced aspect-level sentiment analysis models as the compared methods, the details of which are as follows:

Target Dependency Graph Attention network model (TD-GAT) [33]: First, the model uses syntactic dependency relationships between words to construct a dependency graph and capture contextual information about aspect words. To make explicit use of the target information in GAT, the model uses the LSTM model after GAT to model the cross-layer dependency relationship of aspect words. The model trains word embedding vectors using the Glove model.

Aspect-specific Graph Convolutional Network over Dependency Trees (ASGCN-DT) [21]: This is the first model to solve the problem of aspect-level sentiment classification using GCN. The model proposes the assumption that the dependency relationship between words is asymmetrical, i.e., the constructed graph is a digraph. Specifically, the model constructs a GCN on top of the LSTM and uses a masking mechanism to filter out non-aspect words.

[2] https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools
[3] https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools
[4] https://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools
[5] https://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools

Aspect-specific Graph Convolutional Network over Dependency Graphs (ASGCN-DG) [21]: It is the same as ASGCN-DT, but the dependency relationship between words is symmetric, that is, the constructed graph is undirected, and each word has a dependency relationship with itself. The adjacency matrix of ASGCN-DT is sparser than that of ASGCN-DG.

Autoregressive Feature-based Graph Convolutional Network (AFGCN) [34]: A graph perception model based on interactive GCN. The feature of this model lies in that it does not capture the semantic relations between words, but only uses the relations between aspect words to predict. Specifically, the model uses the Glove model to train word embedding vectors and then builds a full connection graph for aspect words. The GCN network is used on the connection graph for sentiment prediction.

BERT [35]: A word embedding vector model. This model adds the sentence beginning flag "[CLS]" and the sentence end flag "[SEP]" to the input sentence text.

Attentional Encoder Network (AEN)+BERT [36]: The AEN model is an encoder based on an attentional mechanism. Specifically, the attention encoder first perceives semantic information using a multi-head attention mechanism different from the Transformer and then transforms contextual information using Point-wise Convolution Transformation (PCT). This model improves the problem that RNN cannot parallelize and capture long-distance semantic information and introduces a label smoothing regularization mechanism to solve the problem of label unreliability.

Selective Attention Based Graph Convolutional Networks (SA-GCN)+BERT [37]: A GCN model based on the selective attention mechanism, which is used to capture the opinion words corresponding to aspect words in the deep GCN model and improve the accuracy of sentiment analysis. This model uses the self-attention mechanism to select k words with the highest attention scores and then generate the top k attention graph, on which another GCN model is applied to integrate contextual word information.

Sentiment Dependencies with Graph Convolutional Networks (SDGCN)+BERT [23]: A bidirectional attention mechanism with position coding is proposed to model each aspect word and its context word. Specifically, the model uses the GCN model on the attention mechanism and uses position encoding to capture the sentiment dependence between different aspects.

GraphMerge [38]: To avoid errors in the syntactic parser, the model integrates the predicted results of different parsers. Specifically, the model first proposes a way to be able to aggregate the results of different parsers and then builds GNN on the results. The model avoids over-parameterization and overfitting of the GNN layer stacks by introducing more connectivity into the integration graph.

AFGCN + BERT [39]: Using BERT to train the AFGCN model.

4.2.3 Experiment Results and Analysis

The results of the ASGCN model validity experiment are shown in Table 3.

It can be seen from Table 3 that the ASGCN model proposed in this paper has the best performance in all datasets. From the longitudinal point of view, the prediction effect of TD-GAT, ASGCN-DT, ASGCNDG, and AFGCN with the Glove model is not as good as that of the BERT model. Horizontally, the ASGCN model has the highest accuracy on Rest16 and the highest F1 score on Rest14. Our ASGCN takes the differences between syntactic dependency types into account. Meanwhile, the relationship between aspect words is involved in the modeling process. ASGCN therefore can capture richer semantics in sentences and achieve high performance in the experiments.

Table 3: Experimental results of ASGCN compared with baseline model on four datasets

Models	Rest14		Lap14		Rest15		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TD-GAT	81.10	——	73.70	——	——	——	——	——
ASGCN-DT	80.86	72.19	74.14	69.24	79.34	60.78	88.69	66.64
ASGCN-DG	80.77	72.02	75.55	71.05	79.89	61.89	88.99	67.48
AFGCN	81.79	74.42	76.96	73.29	81.55	65.08	89.12	70.60
BERT	84.11	76.68	77.59	73.28	83.48	66.18	90.10	74.16
AEN+BERT	83.12	73.76	79.93	76.31	——	——	——	——
SA-GCN+BERT	85.80	79.70	81.70	78.80	——	——	——	——
SDGCN+BERT	83.57	76.74	81.35	78.34	——	——	——	——
GraphMerge	85.16	77.91	80.00	76.50	——	——	——	——
AFGCN+BERT	85.37	79.26	81.53	78.10	84.02	69.89	89.76	75.88
ASGCN	**86.34**	**79.96**	**81.75**	**79.12**	**84.30**	**70.05**	**90.15**	**76.12**

4.3 Validity Experiment of U-ASGCN Model

4.3.1 Dataset Construction

The dataset used in the current aspect-level sentiment classification research only includes texts, aspect-level words, and labels, but lacks necessary user information. The BiLSTM-CRF model and the ASGCN model mentioned in this paper are applied to construct a dataset containing user information named the user-aspect dataset. The dataset is constructed from the Yelp public dataset. This work is the key point as well as the difficulty of this study. So far, this dataset is the only aspect-level sentiment analysis dataset that contains user information. The construction process of the dataset is as follows.

Step 1: Filter the important data items. The Yelp dataset contains five JSON files, which are user data, product information, comment data, shop registration information, and short comment data. It is worth noting that although sentiment labels of the comment text are retained here, sentiment labels are not substantially involved in the process of the dataset construction.

Step 2: Extract aspect words. In this paper, the BiLSTM-CRF model is used to extract aspect words from the comment text. Specifically, since the comment text in the dataset is a document containing multiple sentences, the comment text needs to be divided into segments first. After extraction, "| |" is used to separate various aspects of the same sentence.

Step 3: Label the sentiment polarity. That is, use the ASGCN model proposed in this paper to label the aspect words extracted in Step 2. Different sentiment labels of the same sentence are separated with "| |".

Step 4: Manually adjust labels to reduce errors. After Step 3, the User-aspect dataset takes shape. To avoid errors, the method of manual adjustment is adopted to correct errors based on automatic labeling results. The adjustment is strictly implemented following accordance with the aspect word definition, and the adjustment of sentiment labels will refer to the original sentiment label of the comment text. The sentiment label of the User-aspect dataset is three-dimensional ({-1, 1, 0}) while the sentiment label of the Yelp dataset is 5-dimensional ({1, 2, 3, 4, 5}). To avoid getting caught up in the subjectivity when adjusting labels, Step 4 is carried out by three people. The final label takes the average of the three people's results. The detailed settings of the dataset constructed from the above four steps are shown in Table 4.

Table 4: The settings of user-aspect dataset

User-aspect dataset settings		Quantity
Number of aspect words	Number of positive sentiment words	1774
	Number of negative sentiment words	494
	Number of neural sentiment words	322
	Total	2590
Other	Number of users	150
	Number of comments	1564
	Average length of sentences	16.5
	Average number of aspect words in a sentence	1.7

4.3.2 Experimental Parameter Setting

The parameter settings of the experiment are shown in Table 5.

Table 5: Experimental parameter settings of U-ASGCN model validity

Parameter name	Value	Meaning
Lr	9×10^{-5}	Learning rate of optimizer
Dropout	0	Probability of dropout layer
l2reg	5×10^{-6}	Optimizer L2 regularization
Batch_size	16	Number of sentences contained in the batch
Num_epoch	15	Number of rounds of training
Max_seq_len	100	Maximum length of the sentence

4.3.3 Bseline Model

The ASGCN with random user feature vectors: The user feature vector matrix is randomly initialized and constantly updated during model training. The user feature vector in this model is embedded after the SentenceGCN sub-model. The purpose of using this model as a baseline model is to verify the necessity of learning the user features beforehand.

Non-adaptive adjustment method: In this method, the user feature vector is embedded after the SentenceGCN sub-model, as shown in Eq. (31), but $\alpha 3$ is fixed at 0.3 in Eq. (31).

Embed before AspectsGCN: This method, as shown in embedding Method 1, can embed user features throughout the calculation process of the model.

Embed after SentenceGCN: This method, as shown in embedding Method 2, can retain complete user information and text information during the prediction phase.

4.3.4 Experimental Results and Analysis

The experimental results are shown in Table 6.

Table 6: Experimental results of user features embedding methods

Model		Acc	F1 score
ASGCN		80.89	57.60
The ASGCN with random user feature vectors		82.24	65.45
Embedded ASGCN model based on user features	Non-adaptive adjustment	82.24	66.08
	Embed before AspectsGCN	79.15	63.69
	Embed after SentenceGCN	83.40	71.29

As can be seen from Table 6, on the User-aspect dataset, the effect of three models with embedded user features is better than that of the ASGCN model without user features embedded. The reason is that the feature of the historical comments of a user contributes to the sentiment classification besides contextual terms. U-ASGCN considers the relationship between the user feature extraction and contextual terms, hence it outperforms ASGCN.

Compared with the ASGCN model with random user feature vectors, two of the three embedding methods are better than this model, namely, the non-adaptive adjustment method and embedding after SentenceGCN. Among them, embedding after SentenceGCN improves the most. This shows that the integrity of the text information is very important for sentiment classification, and the early introduction of other types of data into the model will reduce the accuracy.

Further comparison of the effect between the method of embedding after SentenceGCN and the non-adaptive adjustment method shows that the method of embedding after SentenceGCN has better performance, with an increase of 1.16 percentage points in accuracy and 5.21 percentage points in F1 score. It is speculated that the method of embedding after SentenceGCN is constantly updated along with the learning process of the model, which can more accurately calculate how many user features need to be embedded in the text. This shows that adaptive embedding of user features is more effective.

5 Conclusion

Deep learning methods have been widely used in many fields in recent years, including sentiment analysis. Deep learning methods such as GCN have improved the performance of sentiment analysis models. In this paper, a deep learning method named the U-ASGCN model is proposed to integrate user information and text information. Specifically, the model consists of two sub-models, UCFE and ASGCN, which can use the powerful capabilities of deep learning networks to fully extract features and learn context, including extracting common features of users' historical comments and focusing on the semantic relationships between terms and types of dependencies between words. Experimental results on two Yelp datasets show that the user features extracted by the UCFE model are effective. The experimental results on four SemEval datasets show that the ASGCN model has better predictive performance than the baseline model. Due to the lack of datasets that meet the experimental requirements, this paper constructs the first aspect-level sentiment analysis dataset containing user information. The experimental results on this dataset show that the U-ASGCN model is effective and that embedding user features in the prediction stage can improve the model performance. The proposed scheme is only applicable to the text data. In future work, we will extend U-ASGCN to the task of multimodal sentiment analysis [40].

Acknowledgement: Thanks for the reviewer's comments on the paper.

Funding Statement: This work is partly supported by the Fundamental Research Funds for the Central Universities (CUC230A013). It is partly supported by Natural Science Foundation of Beijing Municipality (No. 4222038). It is also supported by National Natural Science Foundation of China (Grant No. 62176240).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Chumeng Zhang, Jiazhao Chai; data collection: Jianxiang Cao; analysis and interpretation of results: Chumeng Zhang, Jiazhao Chai, Jianxiang Cao; draft manuscript preparation: Chumeng Zhang, Jiazhao Chai, Jialing Ji, Tong Yi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors. The data that support the findings of this study are available from the corresponding author, Tong Yi, upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015. doi: 10.1038/nature14539.

[2] Y. Zhou, Z. Yang, X. Zhang, and Y. Wang, "A hybrid attention based deep neural network for simultaneous multi-sensor pruning and human activity recognition," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25363–25372, Aug. 2022. doi: 10.1109/JIOT.2022.3196170.

[3] A. Manickam, J. Jiang, Y. Zhou, A. Sagar, R. Soundrapandiyan and R. D. J. Samuel, "Automated pneumonia detection on chest X-ray images: A deep learning approach with different optimizers and transfer learning architectures," *Meas.: J. Int. Meas. Confed.*, vol. 184, pp. 109953, 2021. doi: 10.1016/j.measurement.2021.109953.

[4] C. Chen, K. L. Li, S. G. Teo, X. Zou, K. Li and Z. Zeng, "Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 4, pp. 1–42, May 2020. doi: 10.1145/3385414.

[5] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012. doi: 10.1007/978-3-031-02145-9.

[6] J. Tao and L. Zhou, "A weakly supervised wordnet-guided deep learning approach to extracting aspect terms from online comments," *ACM Trans. Manag. Inf. Syst.*, vol. 11, no. 3, pp. 1–22, Jul. 2020. doi: 10.1145/3399630.

[7] B. Marouane, K. Mohammed, and B. H. Abderrahim, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl. Based Syst.*, vol. 226, pp. 107134, May. 2021. doi: 10.1016/j.knosys.2021.107134.

[8] X. Ye, J. Cao, F. Xu, H. Guo, and L. Yin, "Adaptive learning method of Chinese domain sentiment dictionary," *Comut. Eng. Design*, vol. 41, no. 8, pp. 2231–2237, 2020.

[9] P. K. Jain, W. Quamer, V. Saravana, and R. Pamula, “Employing BERT-DCNN with sentic knowledge base for social media sentiment analysis,” *J. Ambient Intell. Human Comput.*, vol. 14, no. 8, pp. 1–13, Jan. 2022. doi: 10.1007/s12652-022-03698-z.

[10] C. H. Wu, F. Z. Wu, S. X. Wu, Z. G. Yuan, and Y. F. Huang, “A hybrid unsupervised method for aspect term and opinion target extraction,” *Knowl. Based Syst.*, vol. 148, no. 1, pp. 66–73, Jan. 2018. doi: 10.1016/j.knosys.2018.01.019.

[11] J. Wagner *et al.*, “DCU: Aspect-based polarity classification for SemEval Task 4,” in *Proc. SemEval*, Dublin, Ireland, 2014, pp. 223–229.

[12] M. S. Akhtar, D. Gupta, A. Ekbal, and P. Bhattacharyya, “Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis,” *Knowl. Based Syst.*, vol. 125, no. 1–2, pp. 116–135, Mar. 2017. doi: 10.1016/j.knosys.2017.03.020.

[13] Y. Wang, Q. Chen, M. Ahmed, Z. Li, W. Pan and H. Liu, “Joint inference for aspect-level sentiment analysis by deep neural networks and linguistic hints,” *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2002–2014, May. 2021. doi: 10.1109/TKDE.2019.2947587.

[14] S. L. Ramaswamy and J. Chinnappan, “RecogNet-LSTM+CNN: A hybrid network with attention mechanism for aspect categorization and sentiment classification,” *J. Intell. Inf. Syst.*, vol. 58, no. 2, pp. 379–404, Jan. 2022. doi: 10.1007/s10844-021-00692-3.

[15] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, “ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis,” *Future Gener. Comput. Syst.*, vol. 115, no. 3, pp. 279–294, Sept. 2021. doi: 10.1016/j.future.2020.08.005.

[16] W. Zhong, D. Tang, J. Wang, J. Yin, and N. Duan, “UserAdapter: Few-shot user learning in sentiment analysis,” in *Proc. ACL-IJCNLP*, Bangkok, Thailand, 2021, pp. 1484–1488.

[17] F. Mireshghallah, V. Shrivastava, M. Shokouhi, T. Berg-Kirkpatrick, R. Sim and D. Dimitriadis, “UserIdentifier: Implicit user representations for simple and effective personalized sentiment analysis,” in *Proc. 2022 Conf. North American Chap. Assoc. Comput. Linguis.: Human Lang. Technol.*, Seattle, USA, 2022, pp. 3449–3456.

[18] C. Stelzmüller, S. Tanzer, and M. Schedl, “Cross-city analysis of location-based sentiment in user-generated text,” in *Proc. Web Conf. 2021 (WWW’21)*, Ljubljana, Slovenia, 2021, pp. 339–346.

[19] A. Mahmoudi, “Identifying biased users in online social networks to enhance the accuracy of sentiment analysis: A user behavior-based approach,” arXiv:2105.05950v1, 2021.

[20] A. Mahmoudi, V. W. Yeung, and W. K. Eric, “User behavior discovery in the COVID-19 era through the sentiment analysis of user tweet texts,” arXiv:2104.08867v1, 2021.

[21] C. Zhang, Q. C. Li, and D. W. Song, “Aspect-based sentiment classification with aspect-specific graph convolutional networks,” in *Proc. EMNLP-IJCNLP*, Hong Kong, China, 2019, pp. 4568–4578.

[22] A. P. B. Veyseh, N. Nouri, F. Dernoncourt, Q. H. Tran, D. J. Dou and T. H. Nguyen, “Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation,” in *Proc. EMNLP*, Punta Cana, Dominican Republic, 2020, pp. 4543–4548.

[23] P. L. Zhao, L. L. Hou, and O. Wu, “Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification,” *Knowl. Based Syst.*, vol. 193, no. 6, pp. 105443, April. 2020. doi: 10.1016/j.knosys.2019.105443.

[24] K. Wang, W. Z. Shen, Y. Y. Yang, X. J. Quan, and R. Wang, “Relational graph attention network for aspect-based sentiment analysis,” in *Proc. ACL*, Seattle, USA, 2020, pp. 3229–3238.

[25] Y. H. Tian, G. M. Chen, and Y. Song, “Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble,” in *Proc. NAACL-HLT*, 2021, pp. 2910–2922.

[26] R. F. Li, H. Chen, F. X. Feng, Z. Y. Ma, X. J. Wang and E. Hovy, “Dual graph convolutional networks for aspect-based sentiment analysis,” in *Proc. ACL/IJCNLP*, vol. 1, 2021, pp. 6319–6329. doi: 10.18653/v1/2021.acl-long.

[27] F. Y. Meng, J. L. Feng, D. P. Yin, S. Chen, and M. Hu, “A architecture-enhanced graph convolutional network for sentiment analysis,” in *Proc. EMNLP*, Punta Cana, Dominican Republic, Nov. 2020, pp. 586–595.

[28] L. Bin, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowl. Based Syst.*, vol. 235, no. 4, pp. 586–595, Jan. 2022. doi: 10.1016/j.knosys.2021.107643.

[29] D. Tang, B. Qin, T. Liu, and Y. Yang, "User modeling with neural network for comment rating prediction," in *Proc. IJCAI*, Buenos Aires, Argentina, 2015, pp. 1340–1346.

[30] Z. Wu, X. Dai, C. Yin, S. Huang, and J. Chen, "Improving comment representations with user attention and product attention for sentiment classification," in *Proc. AAAI-18*, New Orleans, LA, USA, 2018, pp. 5989–5996.

[31] R. K. Amplayo, "Rethinking attribute representation and injection for sentiment classification," in *Proc. EMNLP-IJCNLP*, Hong Kong, China, 2019, pp. 5602–5613.

[32] Z. Yuan, F. Wu, J. Liu, C. Wu, Y. Huang and X. Xie, "Neural review rating prediction with user and product memory," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.(CIKM)*, Beijing, China, 2019, pp. 2341–2344.

[33] B. X. Huang and K. Carley, "Syntax-aware aspect level sentiment classification with graph attention networks," in *Proc. EMNLP-IJCNLP*, Hong Kong, China, 2019, pp. 5469–5477.

[34] Y. Liu, X. J. Huang, A. An, and X. H. Yu, "ARSA: A sentiment-aware model for predicting sales performance using blogs," in *Proc. SIGIR*, New York, NY, USA, 2007, pp. 607–614.

[35] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, USA, vol. 1, 2019, pp. 4171–4186.

[36] Y. W. Song, J. H. Wang, T. Jiang, Z. Y. Liu, and Y. H. Rao, "Attentional encoder network for targeted sentiment classification," in *Proc. 28th Int. Conf. Artif. Neural Netw.(ICANN)*, Munich, Germany, 2019, pp. 93–103.

[37] X. C. Hou, J. Huang, G. T. Wang, K. Huang, X. D. He and B. Zhou, "Selective attention based graph convolutional networks for aspect-level sentiment classification," in *Proc. Fifteenth Workshop Graph-Based Methods Nat. Lang. Process (TextGraphs-15)*, Mexico City, Mexico, 2019, pp. 83–93.

[38] X. C. Hou *et al.*, "Graph ensemble learning over multiple dependency trees for aspect-level sentiment classification," in *Proc. NAACL-HLT*, 2021, pp. 2884–2894.

[39] B. Liang, R. D. Yin, L. Gui, J. C. Du, and R. F. Xu, "Jointly learning aspect-focused and inter-aspect relations with graph convolutional networks for aspect sentiment analysis," in *Proc. Coling*, Barcelona, Spain, 2020, pp. 150–161.

[40] Y. Li, K. Zhang, J. Wang, and X. Gao, "A cognitive brain model for multimodal sentiment analysis based on attention neural networks," *Neurocomputing*, vol. 430, no. 2, pp. 159–173, Oct. 2021. doi: 10.1016/j.neucom.2020.10.021.