Contents lists available at ScienceDirect

Knowledge-Based Systems

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

CABiLSTM-BERT: Aspect-based sentiment analysis model based on deep implicit feature extraction

Bo He, Ruoyu Zhao *, Dali Tang

College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

ARTICLE INFO

Keywords:
Aspect-based sentiment analysis
Natural language processing
BERT
Transfer learning
Sentiment analysis

ABSTRACT

Aspect-based sentiment analysis (ABSA) models typically focus on learning contextual syntactic information and dependency relations. However, these models often struggle with losing or forgetting implicit feature information from shallow and intermediate layers during the learning process, potentially compromising classification performance. We consider the implicit feature information in each layer of the model to be equally important for processing. So, this paper proposes the CABiLSTM-BERT model, which aims to fully leverage implicit features at each layer to address this information loss problem and improve accuracy. The CABiLSTM-BERT model employs a frozen BERT pre-trained model to extract text word vector features, reducing overfitting and accelerating training. These word vectors are then processed through CABiLSTM, which preserves implicit feature representations of input sequences and LSTMs in each direction and layer. The model applies convolution to merge all features into a set of embedding representations after highlighting important features through multi-head self-attention calculations for each feature group. This approach minimizes information loss and maximizes utilization of important implicit feature information at each layer. Finally, the feature representations undergo average pooling before passing through the sentiment classification layer for polarity prediction. The effectiveness of the CABiLSTM-BERT model is validated using five publicly available real-world datasets and evaluated using metrics such as accuracy and Macro-F1. Results demonstrate the model's efficacy in addressing ABSA tasks.

1. Introduction

Current sentiment analysis (SA) can be categorized into three main types: chapter-based sentiment analysis, sentence-based sentiment analysis, and aspect-based sentiment analysis (ABSA) [1]. Traditional sentiment analysis is mainly chapter-level sentiment analysis and sentence-level sentiment analysis, in which the overall analysis of the whole text is chapter-level sentiment analysis, while the analysis of the whole sentence is sentence-level sentiment analysis, both of which can only analyze the overall sentiment of the text or sentence, and cannot analyze multiple aspect-based sentiments, so they are classified as coarse-grained sentiment analysis.

In contrast, aspect-based sentiment analysis is able to analyze the sentiment of specific aspects of a sentence rather than the overall sentiment, and is a fine-grained sentiment analysis [2]. For example, "This graphics card works well, but the graphics memory is too small." (Fig. 1) This sentence has two opposite sentiment polarities, positive and negative, with "graphics card" being positive and "graphics memory" being negative. In this case, it is difficult for the coarse-grained chapter-based sentiment analysis and sentence-based sentiment analysis methods to accurately analyze the aspect-based sentiment of the text. Therefore, the demand for sentiment analysis in today's complex semantic environment can no longer be met by traditional coarse-grained sentiment analysis methods [3]. And aspect-based sentiment analysis, a fine-grained sentiment analysis approach, has received increasing attention in the past decade and has become one of the important research topics in the field of natural language processing [4].

Earlier, in order to be able to capture feature representations relatively easily, researchers mainly used methods such as machine learning-based and sentiment lexicon-based approaches for text aspect-based sentiment analysis [5,6]. However, the input text must undergo extensive preprocessing as well as complex feature engineering before using these methods, and the quality of the manually labeled data largely affects the effectiveness of the results, which requires a large investment in labor costs when dealing with the massive amount of text data nowadays. In addition, these models do not or less take into account

* Corresponding author.
E-mail addresses: hebo@cqut.edu.cn (B. He), qiaqiaeric@outlook.com (R. Zhao).

https://doi.org/10.1016/j.knosys.2024.112782
Received 4 March 2024; Received in revised form 11 October 2024; Accepted 21 November 2024
Available online 23 November 2024

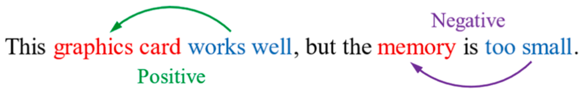

Fig. 1. An example of multi sentiment polarities sentence.

the semantic relations within the utterance, contextual relations and other elements.

Deep learning has become more and more mature and more and more scholars in the field of natural language processing (nlp) have started to use neural networks to handle tasks [7]. In fact deep learning approaches are still the latest and most effective techniques for with aspect-based sentiment analysis [8]. Deep learning based neural networks are able to convert text into entity vectors using word embeddings thus overcoming the problem of traditional machine learning that relies on manual extraction of text features. Due to the advantage of being able to easily capture sequential and contextual semantic information of text, a large number of researchers have used Recurrent Neural Networks (RNN) and attention mechanisms in aspect-based sentiment analysis tasks [9]. However, when multiple pieces of sentiment information are in the same text, aspect-based sentiment analysis requires the model to be able to capture and understand deeper and more subtle sentiment information compared to sentence-based or chapter-based sentiment analysis [10]. In this case, models that can only simply handle text order information and contextual relationships are less adequate.

In this context, using pre-training models to enhance the semantic representation and thus deepen the model's learning of the syntactic dependencies of the text is a good approach. The "Embeddings from Language models (Elmo)" [11] and "Generative pre-training (GPT)" [12] models, which show very good performance, and the subsequent emergence of the BERT model [13] also performed well, which prompted us to decide to incorporate BERT into our model for extracting shallow features from text. In fact the emergence of pre-trained language models (PLMs) such as BERT has brought substantial improvements to the ABSA task in recent years. The generalization ability and robustness of PLM-based ABSA models have been significantly improved [14].

In Karimi et al.'s paper [15], they used the standard BERT model for the ABSA processing task and found that the performance of BERT can easily outperform the general neural network model. Despite the excellent performance of the BERT model, the large number of parameters leads to slow convergence during model training. It is very prone to overfitting when the dataset is not sufficiently large, and using BERT alone is not enough to enhance the model's performance. To mitigate the overfitting issue in the model and to enhance its performance, we experiment with freezing the BERT model and adding layers tailored for the ABSA task, as an alternative to fine-tuning BERT. Following the extraction of shallow implicit semantic features from the text using a pre-trained BERT model, we implement our proposed CABiLSTM architecture to capture more profound implicit semantic features. Simultaneously, this approach facilitates the fusion of multi-level implicit features, resulting in a comprehensive set of information-rich implicit representations. The CABiLSTM framework enables us to perform deep extraction of implicit feature information while significantly mitigating the loss of crucial feature data. Consequently, this methodology enhances the model's recognition accuracy by leveraging implicit features to their full potential.This paper introduces CABiLSTM, a novel approach aimed at enhancing model performance by fully leveraging implicit features extracted from various layers. The primary objective is to improve classification accuracy by maximally retaining and utilizing key implicit features of the text post-extraction. We propose an integrated CABiLSTM-BERT model, which employs a pre-trained BERT with frozen weights as an initial feature extractor. This strategy not only accelerates the training process but also minimizes the risk of overfitting.

In the CABiLSTM architecture, the shallow features extracted by BERT undergo processing through multiple BiLSTM layers, resulting in diverse sets of implicit features with varying directionality and depth. Drawing inspiration from [11], we recognize that each of these feature sets encapsulates unique information. Consequently, we preserve the outputs from each layer and direction independently, thereby enabling the model to harness a broader spectrum of information.

Subsequently, multi-head self-attention mechanisms are applied to each set of implicit features. This step serves to accentuate salient features while simultaneously mitigating the impact of noise. The resulting feature sets are then concatenated along the channel dimension and merged through convolutional operations, yielding an output rich in diverse and significant implicit feature values.

For sentiment polarity classification, the model employs a softmax function to compute probability distributions. This novel approach significantly enhances the classification accuracy in ABSA task.

To validate the efficacy of our proposed model, we conducted comprehensive experiments across five distinct datasets. The results demonstrate the superior performance of CABiLSTM-BERT in ABSA task.

The main contributions of this paper are as follows:

- We propose a novel approach to aspect-based sentiment classification that leverages multi-layer feature extraction and fusion, it takes the novel perspective of fully utilizing the implicit feature information extracted from each layer to improve the performance of the model on the ABSA task. The key innovation of our method lies in its ability to retain and integrate implicit features from shallow to deep layers. By doing so, our approach mitigates information loss during the model's layer-by-layer data processing and minimizes the neglect of detailed information, thereby significantly improving classification accuracy.
- Leveraging the concept of fully utilizing implicit feature information, we developed the CABiLSTM-BERT model. This architecture integrates Bidirectional Long Short-Term Memory (BiLSTM), multi-head self-attention, and multi-channel convolution to construct the CABiLSTM component. Additionally, we employed a pre-trained BERT model with frozen weights as an efficient word embedding extractor. By extracting features layer by layer and then combining them, it can effectively extract and retain important implicit feature information from the text, thus improving classification accuracy by reducing information loss.
- We conducted extensive experiments on five public datasets and demonstrated that the CABiLSTM-BERT can effectively improve classification accuracy by extracting deep features as well as reducing feature loss to achieve excellent performance.

The rest of this study is as follows: Section 2 introduces related work on aspect-based sentiment analysis; Section 3 provides a detailed description of the proposed model; Section 4 presents the experiments conducted to validate the effectiveness of this model; Section 5 examines the effects of model architecture modifications and evaluates the pros and cons of categorical balance in model training; Section 6 summarizes the work and outlines our future directions.

2. Related work

This study centers on aspect-based sentiment classification within the domain of aspect-based sentiment analysis. Subsequent references to aspect-based sentiment analysis will pertain specifically to aspect-based sentiment classification. Aspect-based sentiment analysis represents a more nuanced subset within the category of sentiment analysis tasks. In contrast to assessing the overall sentiment polarity of entire passages or sentences, the objective of aspect-based sentiment analysis is to identify the sentiment polarity associated with particular entities or aspect terms within a sentence or text.

In early research, aspect-based sentiment analysis tasks were mainly based on traditional machine learning models [16], rule-based [17], sentiment lexicon [18] and bag-of-words models [19]. For example, Zheng et al. [20] combined Word Frequency-Inverse Document Frequency (TF-IDF) and Support Vector Machines (SVMs) to determine sentiment tendencies of review texts. Jun Liu et al. extended the CBoW (Continuous Bag of Words) word vector model to propose a cross-domain emotion-aware word embedding learning model that captures both emotional information and domain relevance of words [21].

Currently, the approaches for aspect-based sentiment analysis tasks predominantly involve traditional machine learning methods and deep learning methods. Among these, deep learning-based methods are increasingly becoming the mainstream for aspect-based sentiment analysis tasks [22].

In recent years, Recurrent Neural Networks (RNN), as an end-to-end neural network model, have demonstrated excellent automatic feature extraction and classification in aspect-based sentiment analysis [23]. In order to incorporate sentiment information into document-level data for ABSA modeling, Li et al. [24] proposed a framework called Semi-Supervised and Multi-Task Learning (SEML), which uses embedding and LSTM layers. Meanwhile, Chen and Qian [25] developed a method called TransCap for sharing document-level prior knowledge for aspect-based sentiment analysis tasks. In addition, Su et al. [26] proposed an ABSA model based on capsule networks and XLNet that captures the relationship between sequences and aspects and improves aspect awareness and XLNet pre-training to handle task uncertainty. For the Twitter comment sentiment classification problem, Dong et al. [27] proposed a model called Adaptive Recurrent Neural Network (AdaRNN).

However, these models may ignore the implicit features of intermediate computations when using recurrent neural networks. In contrast, the Elmo [11] model effectively preserves the implicit features computed by each layer of the recurrent neural network to learn deep contextual word representations. However, the way the Elmo [11] model merges the features of each layer may ignore the key features of certain layers. To address this problem, this study improves the way the Elmo [11] model merges features. Specifically, we first use the multi-head self-attention mechanism to highlight important features in each layer, and then utilize convolution to merge the data from each layer in order to retain important implicit features and reduce the effect of noise, enhancing the model's ability to understand the input text.

The emergence of the Transformer architecture [28] has led to a significant increase in the number of models that adopt attention mechanisms as their core structure. Zhao et al. [29] proposed an attention-guided fusion model, known as the Attention Transfer Network (ATN), which utilizes the attention weights from document-level sentiment analysis models as learning cues, thereby adapting the model to the ABSA task during the training process. Additionally, pre-trained language models such as GPT [12] and BERT [13] have excelled in a wide range of text processing tasks, demonstrating exceptional performance. Nonetheless, these pre-trained models are not without their limitations; their extensive parameter counts result in lengthy training durations and a heightened risk of overfitting.

Our model incorporates BERT, but only uses it as a word vector feature extractor, and freezes the weights of BERT using transfer learning. This approach effectively reduces training time and mitigates the impact of overfitting.

Additionally, various researchers have made enhancements to neural networks for ABSA tasks and achieved promising results. For instance, Sun et al. [30] proposed a Dependency Tree-based GCN model to enhance the feature representations of aspects in Bi-directional Long and Short-Term Memory (Bi-LSTM) learning, yielding significant improvements. Liang et al. [31] integrated sentiment knowledge from SenticNet to construct a GCN, effectively enhancing the dependency graph of sentences. Furthermore, Zhang et al. [32] introduced an aspect-aware attention mechanism to compute the attention score matrix of a sentence, which was then utilized as an initial adjacency matrix by the GCN to augment the dependency graph of a sentence based on the syntactic structure of the sentence. This approach improved the GCN's performance in handling the ABSA task by considering the syntactic dependencies between words at different distances. Moreover, Bo Huang et al. [33] proposed a model that combines a conditional random field with a graph convolutional neural network to address the impact of multiple aspectual words on model accuracy in discourse.

Zeng et al. [34] proposed a novel multi-task learning network for relationship construction, which employs a graph convolutional network to encode aspect representations. Phan et al. [35] utilized contextual information, semantic relationships, and syntactic structure simultaneously to construct a graph convolutional network. To enhance the current graph convolutional networks' ability to model sentiment dependencies, Zhao et al. [36] introduced an Aggregate Graph Convolutional Network (AGCN). Feng et al. [37] developed a new model, AG-VSR, which employs variable sentence representations and attention-assisted graph representations generated by GCN for aspect-based sentiment classification.

These models generally focus on syntactic, grammatical, and dependency features, to some extent overlooking the issue of information loss. Our model, through CABiLSTM, is able to minimize information loss while emphasizing features, thereby preserving important features to the greatest extent possible.

Our proposed CABiLSTM-BERT model effectively captures crucial implicit features in the textual data, enhancing the model's comprehension without significant loss of information. Notably, our model demonstrates outstanding accuracy and outperforms existing models on all four datasets utilized in our study, establishing its competitiveness in the field.

3. Proposed model

In this section, we describe in detail our proposed CABiLSTM-BERT model, which consists of a word embedding layer, an implicit feature extraction layer, and a sentiment classification layer. In the whole model, the implicit feature extraction layer plays a central role. Upon examining the model's overall structure, the text data is first encoded and then input into BERT for word vector extraction. Additionally, to reduce overfitting and training time, we adopt a transfer learning approach by freezing the weights of BERT, preventing them from being updated during training. BERT plays a role in extracting word vectors in this process.

Subsequently, the word vectors are fed into CABiLSTM. The word vectors undergo processing through 2 layers of BiLSTM. The feature represent-ations output by each BiLSTM layer, as well as the initially input word vectors (the output of BERT), are preserved and separately used to compute multi-head self-attention values. After being processed by the multi-head self-attention mechanism, the resulting multi-group feature representation is spliced in the channel dimension and subsequently fed into the convolutional layer. The convolutional layer processes the original multi-channel feature representations into a single-channel feature representation, which serves as the output of CABiLSTM. Finally features are fed into the sentiment classification layer to predict sentiment polarity.The overall CABiLSTM-BERT framework is shown in Fig. 2.

3.1. Problem formulation

In order to perform aspect-based sentiment classification, the input to the model requires sentiment aspect words in addition to the original text. Given a sentence with n words, and an aspect in that sentence consists of m words, the sequence can be represented as $S = (w_1, \ldots, w_m, \ldots, w_{m+n})$, where the first m words represent the aspect, and the latter n words represent the original sentence. Prior to the input, S would be processed into $S' = (w_{CLS}, w_1, \ldots, w_m, w_{SEP}, \ldots, w_{m+n}w_{SEP})$. What lies between [CLS] and [SEP] is an aspect word in this sentence, and what lies between [SEP] and [SEP] is the original sentence. In fact to keep the

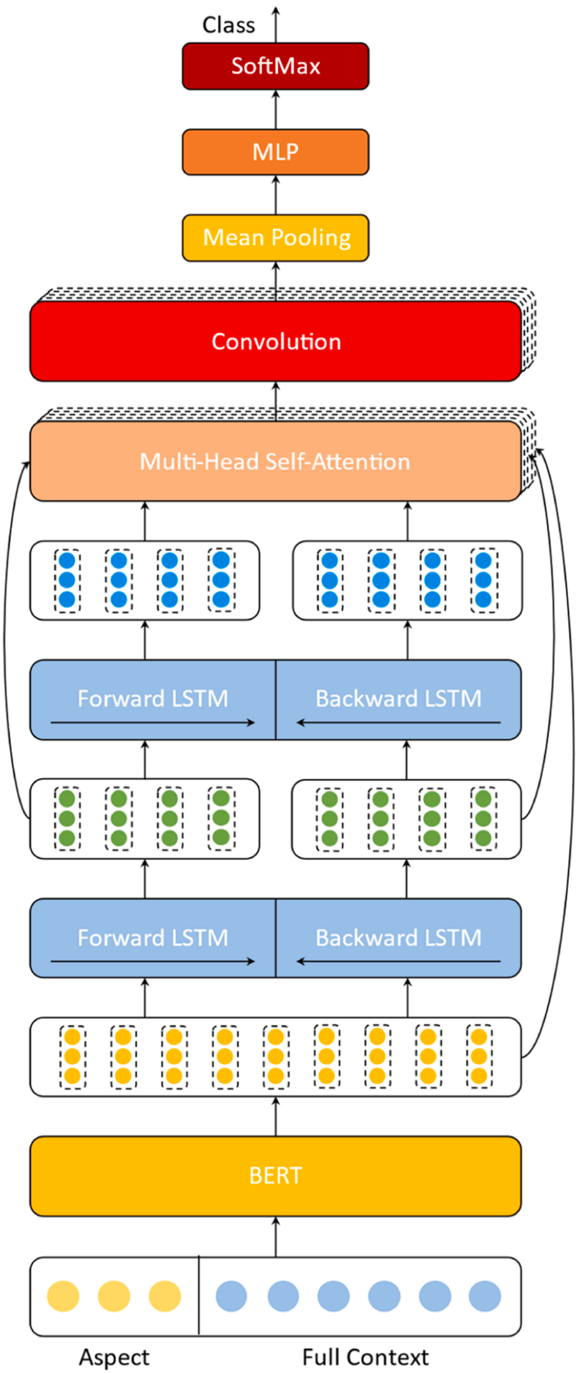

Fig. 2. CABiLSTM-BERT model.

input consistent we will pad the length of S' to 256.

We have established a vocabulary of size 30,522 for the input words. In addition to the word sequence S', we also need an input aspect sequence T and position encoding P to distinguish aspect words from the original sentence. Aspect words and the original sentence are divided into two categories using T, which uses 0, 1 to mark the corresponding positions, $T = (1,\ldots,1,0,\ldots,0)$. P uses integers to represent the positions, $P = (1,2,\ldots,256)$.

Then our S' will be input into BERT for word embedding processing to obtain the word embedding representation Y. When Y enters the CABiLSTM, it will undergo deep privacy feature extraction and be combined into a new feature sequence C_o. Finally, C_o is processed through the sentiment classification layer and the classification result is obtained by fully connected neural network and Softmax function. The key symbols used in this paper have been summarized in Table 1.

3.2. Word embedding layer

The input data sets S', T, and P will be used to calculate the corresponding initial embeddings using the following formulas:

$$E_S = W_s h(S') \tag{1}$$

$$E_T = W_t h(T) \tag{2}$$

$$E_P = W_p h(P) \tag{3}$$

Where E_S is the word embedding representation corresponding to S', E_T is the word embedding representation corresponding to T, E_P is the word embedding representation corresponding to P. $W_s \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $W_t \in \mathbb{R}^{d_{hid} \times d_{hid}}$, and $W_p \in \mathbb{R}^{d_{hid} \times d_{hid}}$ are the corresponding weight matrices, and $h(x)$ denotes the one-hot encoding function.

Next, the corresponding positions of E_S, E_T, and E_P are added together to obtain the encoded vector $X = (x_{CLS}, x_1, \ldots, x_m, x_{SEP}, \ldots, x_{m+n}, x_{SEP})$, as shown in the following formula:

$$X = E_S + E_T + E_P \tag{4}$$

The feature extraction layer relies on BERT [13] to derive word vector representations from the text, with BERT constructed from a series of Transformer encoders. Each encoder consists of multi-head self-attention mechanisms and fully connected layers, and implements residual connections and layer normalization between its layers. These encoders are concatenated to form the BERT model, whose core is a

Table 1
Notation and their definitions.

Notation	Definition
S	Text sequence
m	Aspect words length
n	Original sentence length
w_i	The ith word in S
S'	Input sequence
T	Distinguishing sequence
P	Position coding
E	Word embedding
$h(x)$	One-hot encoding function
X	Encoded output
V	BERT outputs a sequence of word vectors
v_t	Vector of input words at time t
σ	Sigmoid
h_t	Hidden layer state at moment t
f_t	The value of the Forgotten gate
l_t	The value of the memory gate
C'_t	Temporary cellular state
C_t	Current cell state
o_t	Output gate values
out	The amount of data output
$layer$	Number of BiLSTM layers
F_i	Forward LSTM output at layer i
B_i	Backward LSTM output at layer i
Q	Query
K	Key
V	Value
$head_i$	Group V self-attention values
d	The size of the dimension of K
H	Number of heads of attention
A_i	The self-attention value of the ith group of bulls
W_h	Trainable weights when splicing joints
A_o	Spliced multi-channel vector groups
C_o	Convolutional layer output
W_{conv}	Convolutional weights matrix
b_{conv}	Convolutional bias matrix
c_i	The ith word vector
M	The eigenvector after the mean has been calculated

multi-head self-attention mechanism that effectively captures the contextual relationships within the text. BERT is a highly successful pre-trained language model that undergoes pre-training with MLM (Masked Language Model [13]) and NSP (Next Sentence Prediction [13]) tasks before being fine-tuned for specific tasks. The extracted features are then utilized for processing downstream tasks.

However, fine-tuning BERT can be computationally intensive due to its large number of parameters, leading to slow convergence and a higher risk of overfitting when trained on smaller datasets. Consequently, after extensive experimentation, this paper adopts a strategy of freezing BERT's weights, preventing them from updating during training, and uses BERT solely as a feature extractor for word vectors. The text is then encoded as required (Formula. (1)(2)(3)), with BERT processing the text as the corresponding word vector extractor, where the length of the word vectors is set to n and word vector $V = (v_1, \ldots, v_L)$. Training the model in this manner results in faster convergence and achieves commendable results within a reduced timeframe.

3.3. Deep implicit feature extraction layer

Deep implicit feature extraction layer (CABiLSTM) as the core part of the model, the specific structure is shown in Fig. 3. Its role is to deeply extract and retain the implicit features of the text sequence as much as possible, enhancing the model's understanding of the text, and minimize the loss of information. This layer mainly consists of a dual-layer bidirectional LSTM for feature extraction, multi-head self-attention to highlight key points, and a convolutional layer for feature fusion. After the word vectors are input into the implicit feature extraction layer, they undergo processing by the dual-layer BiLSTM, which can capture long-distance dependencies in the text sequence and extract implicit feature representations of the text sequence. In order to minimize the loss of important dependencies and feature representations, the model retains the output of each layer. Furthermore, to highlight important feature values, the model computes the multi-head attention values for each layer and direction separately. Finally, the model uses convolution to merge various vectors to reduce noise and retain important information.

3.3.1. Dual-layer bidirectional long short-term memory network

LSTM [38] is known as Long Short-Term Memory Network, which is sufficient to acquire semantic information in textual contexts and can capture textual dependencies over long distances. The BiLSTM model is composed of the final state vectors of both forward and backward LSTMs linked together, allowing for the extraction of hidden states from both sequences. For our word vector embeddings, we employed a two-layer BiLSTM. The gating unit of the LSTM is essential, and the equations for its different types of gating units are as follows:

The main role of the forgetting gate f_t is to allow the network to forget unimportant features in the memory:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, v_t] + b_f\right) \tag{5}$$

The main role of the memory gate $l_t$is to retain important feature information from the current time step as well as previous time steps:

$$l_t = \sigma(W_l \cdot [h_{t-1}, v_t] + b_l) \tag{6}$$

The temporary cell state C'_t mainly records important information related to the previous time step:

$$C'_t = tanh(W_C \cdot [h_{t-1}, v_t] + b_C) \tag{7}$$

The current moment cell state C_t stores important feature information about the current time and previous time steps:

$$C_t = f_t \times C_{t-1} + l_t \times C'_t \tag{8}$$

The output gate o_t largely retains information about the current time step:

$$o_t = \ \sigma(W_o \cdot [h_{t-1}, v_t] + b_o) \tag{9}$$

The current moment hidden layer state h_t is the output of the current time step, which highlights currently important features based on past contextual information:

$$h_t = o_t \times \tanh(C_t) \tag{10}$$

It should be noted that in the above formulas, the $W_f \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $W_l \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $W_C \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $W_o \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $b_f \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $b_l \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $b_C \in \mathbb{R}^{d_{hid} \times d_{hid}}$, $b_o \in \mathbb{R}^{d_{hid} \times d_{hid}}$ represent the weight matrices and biases

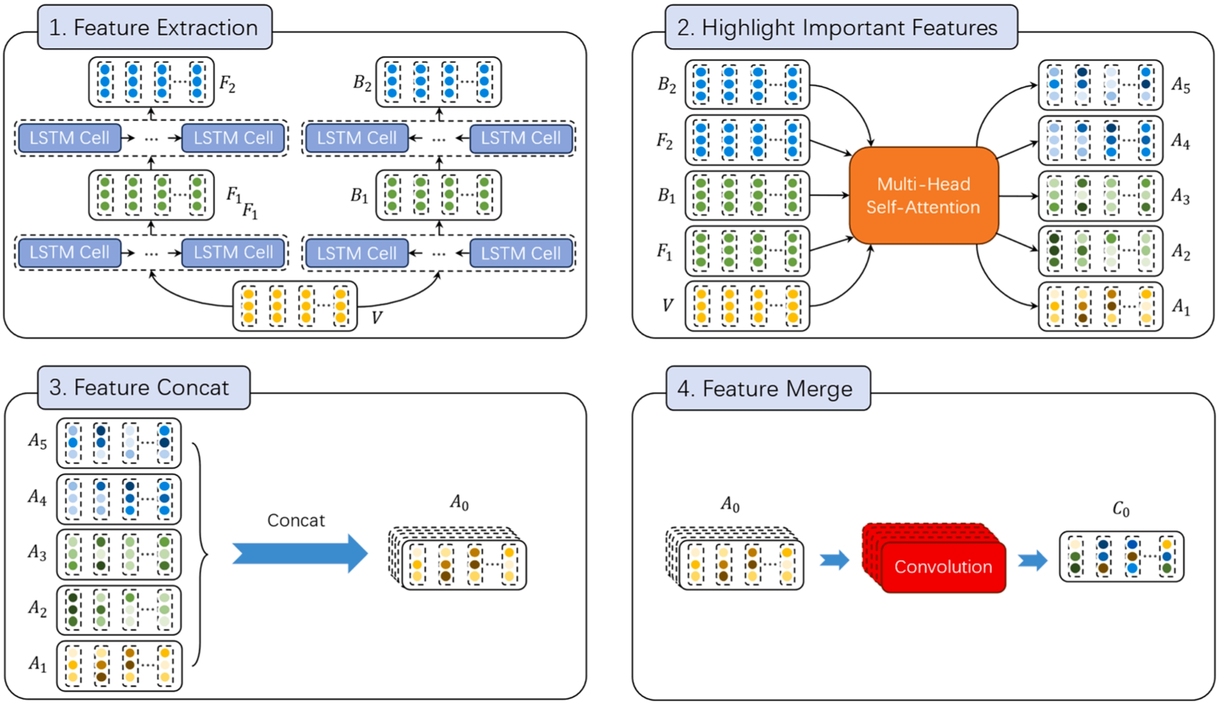

Fig. 3. Deep implicit feature extraction layer (CABiLSTM).

used to compute the corresponding gates or cell states, and $[\alpha,b]$ denotes the concatenation of α and b.

Finally, through the above calculations, we can obtain a sequence $\{h_1,\ldots,h_L\}$ of the same length as the input sequence V. Particularly, the sequence obtained by our BiLSTM calculation does not concatenate the forward and backward sequences of only the last layer, as is the case in traditional BiLSTMs. Instead, it separately retains the forward and backward sequences for each layer of the BiLSTM as $\{h_1,\ \ldots,\ h_L\}$. To describe it clearly, let the forward sequence be denoted as F_i and the backward sequence asB_i, where the index i represents the current layer. Therefore, the output can be represented as $\{V,F_1,B_1,F_2,B_2\}$, as shown in the first part of Fig. 3. It should be noted that the amount of output data varies according to the number of BiLSTM layers, and the formula for calculating the output data amount is as follows:

$$out = layer \times 2 + 1 \tag{11}$$

Where *out* is the output data amount, and *layer* is the number of BiLSTM layers. As the model with better performance uses 2 BiLSTM layers, we will use 2 BiLSTM layers as an example here.

3.3.2. Feature concatenation

In order to identify key features, we will first compute multi-head self-attention (MSA) for $\{V,F_1,B_1,F_2,B_2\}$. Before computing attention, it is necessary to determine *Q*, *K*, *V*. Here, we will use linear layers to process the input into *Q*, *K*, *V*:

$$Q = W_Q Input + b_Q \tag{12}$$

$$K = W_K Input + b_K \tag{13}$$

$$V = W_V Input + b_V \tag{14}$$

Where $W_Q \in \mathbb{R}^{d_Q \times d_{hid}}$, $W_K \in \mathbb{R}^{d_K \times d_{hid}}$, $W_V \in \mathbb{R}^{d_V \times d_{hid}}$, $b_Q \in \mathbb{R}^{d_Q \times d_{hid}}$, $b_K \in \mathbb{R}^{d_K \times d_{hid}}$, $b_V \in \mathbb{R}^{d_V \times d_{hid}}$ are the corresponding weight matrices and biases. Multi-head self-attention is created by multiple parallel heads, each of which employs a self-attention mechanism. The formula for self-attention is as follows:

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{15}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{16}$$

Where $head_i$ represents the self-attention values of the *i*th group of heads. *d* denotes the dimension of *K*, $W_i^Q \in \mathbb{R}^{d_Q \times d_{hid}}$, $W_i^K \in \mathbb{R}^{d_K \times d_{hid}}$, $W_i^V \in \mathbb{R}^{d_V \times d_{hid}}$ are the corresponding weight matrices. Let *H* be the number of multi-head self-attentive heads, then the multi-head attention will be computed by splicing *H* heads as follows:

$$A_i = [head_1, head_2, \ldots, head_H] \cdot W_h \tag{17}$$

Here, A_i represents the *i*th group of multi-head self-attention, and $W_h \in \mathbb{R}^{d_{hid} \times d_{hid}}$ is a trainable weight. Multi-head self-attention fully utilizes the relationships between all words in the entire sequence, regardless of their order and distance. Essentially, it assigns corresponding weights to each word in the sequence based on the inherent feature relationships within the sequence, thereby highlighting important features in the input sequence.

The data $\{V,F_1,B_1,F_2,B_2\}$ obtained from the upper part of the CaBiLSTM is transformed into $\{A_1,A_2,A_3,A_4,A_5\}$ through the calculation of multi-head attention. For each group of data, it is necessary to add a channel dimension. Then, these five groups of data are concatenated along the channel dimension to obtain A_o. As shown in Fig. 4, the data will be transformed into a single-channel feature sequence after convolution. Here, the convolution effectively merges important features and filters out noise, reducing information loss. Specifically, it is calculated by the following equation:

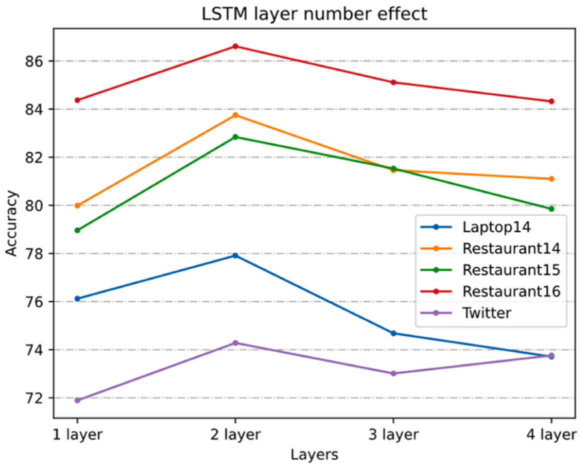

Fig. 4. The impact of the number of LSTM layers.

$$C_o = ReLU(W_{conv} \cdot A_o + b_{conv}) \tag{18}$$

Where C_o is the output of the output of the convolutional layer in the CABiLSTM; $W_{conv} \in \mathbb{R}^{d_{hid} \times d_{hid} \times N_c}$ and $b_{conv} \in \mathbb{R}^{d_{hid} \times d_{hid} \times N_c}$ represent the weights and biases of the convolution.

3.4. Sentiment classification layer

The CABiLSTM layer's output $C_o = \{c_1,\ldots,c_L\}$ is a matrix of shape *equence lenth* × *feature size*, where c_i represents the *i*th word vector. To obtain the classification result from the fully connected layer, the one-dimensional vector of feature size is achieved by averaging over the sequence length dimension. The formula is as follows:

$$M = mean_{seq}(C_o) = \sum_{i=0}^{L} c_i \Big/ L \tag{19}$$

The feature vector M is fed into a fully connected layer for computation. This layer consists of three layers with 1024, 512, and 3 neurons respectively, using the ReLU activation function. The resulting data from the fully connected layer is passed through the Softmax function to determine the probability of belonging to each of the three categories: negative, neutral, and positive. The highest probability corresponds to the polarity of the target aspect in the text. Algorithm 1 shows the algorithm of our proposed model.

4. Experiment

To validate the effectiveness, advancements, and limitations of the proposed model, we conducted a series of experiments, as detailed in this section. First, in Section 4.1, we introduce the five datasets used in these experiments. Following this, Section 4.2 provides a detailed description of the parameter settings, optimizer choices, and training configurations that contribute to the strong performance of the proposed model. In Section 4.3, we discuss the evaluation metrics applied to the Aspect-Based Sentiment Analysis (ABSA) task. Section 4.4 offers a brief overview of several baseline models that are either well-established or recently proposed and have shown promising performance. To demonstrate the effectiveness and advancements of our model, Section 4.5 presents a comparative analysis between our proposed model and the models mentioned in Section 4.4. Finally, in Section 4.6, we conduct ablation studies to further validate the effectiveness of our improvements by systematically removing individual components of the model and analyzing the resulting changes in performance.

Algorithm 1
The proposed algorithm.

```
Input: aspect $A$, sentence $S$
for all epoch in epochs do
    $S', T, P \leftarrow$ BertTokenizer($A, S$)
    $V \leftarrow$ FreezedBert($S', T, P$)
    $F_1 \leftarrow$ ForwardLSTM($V$)
    $B_1 \leftarrow$ BackwardLSTM($V$)
    $F_2 \leftarrow$ ForwardLSTM($F_1$)
    $B_2 \leftarrow$ BackLSTM($B_1$)
    # compute the multi-head self-attention separately
    $(A_1, A_2, A_3, A_4, A_5) \leftarrow$ MultiHeadSelfAttention($V, F_1, B_1, F_2, B_2$)
    $A_0 \leftarrow$ ConcatByChannel($A_1, A_2, A_3, A_4, A_5$)
    $C_0 \leftarrow$ Conv($A_0$)
    $M \leftarrow$ compute by Eq. (19)
    $result \leftarrow$ MLP($M$)
end for
```

4.1. Datasets

We conducted experiments on five publicly available real-world datasets: Lap14, Rest14, Rest15, Rest16, and Twitter. These datasets have been widely used by researchers in the Aspect-Based Sentiment Analysis (ABSA) task. The Lap14 and Rest14 datasets are from Task 4 of SemEval 2014 [39], while the Rest15 dataset is from Task 12 of SemEval 2015 [40]. The Rest16 dataset is from Task 5 of SemEval 2016 [41]. All four of these datasets were constructed by Pontiki et al. In our experiments, we removed instances labeled as "conflict" from these four datasets, retaining only those labeled with polarities of "Positive," "Negative," and "Neutral." The Twitter dataset, on the other hand, consists of tweets compiled by Dong, Wei, Tan, Tang, Zhou, and Xu et al. [42]. Statistical details of these datasets are provided in Table 2.

4.2. Experimental setup

For the CABiLSTM-BERT, our data is first encoded in the format of standard BERT input. The pre-trained weights used by BERT [13] are the bert-base-uncased weights released by HuggingFace. We freeze the weights of BERT during the training of the model so that BERT does not update the weights, which can effectively reduces model overfitting and speeds up training.

The word embedding length and hidden state dimension of both BERT and BiLSTM are set to 256 and 768, respectively. The BiLSTM consists of two layers, where each layer produces forward and backward vectors as part of its output. This differs from the traditional concatenated output. To perform multi-head self-attention, 8 heads are utilized. For the convolution layer, a kernel size of 5 × 5 is employed, along with 5 kernels. The number of kernels varies with the number of BiLSTM layers; however, the models trained in this study, which exhibit superior performance, employ only 2 BiLSTM layers. AdamW [43] optimizer is used with a learning rate of 0.0005 and a batch size of 32. The maximum input text length is set at 256. The proposed CABiLSTM model is implemented using PyTorch.

4.3. Evaluation metrics

To evaluate the model's performance on aspect-based sentiment analysis tasks, we employed two evaluation metrics: accuracy (ACC) and macro-averaged F1 value (Macro-F1). These metrics were used to assess the model's performance accurately.

ACC: Accuracy (ACC) is a metric used to calculate the percentage of samples that a model correctly predicts out of all samples. The formula for calculating the accuracy is as follows:

$$Acc = \frac{T}{N} \tag{20}$$

Where T represents the number of accurately predicted samples, while N denotes the total sample size. A higher accuracy indicates stronger model performance.

Macro-F1: The macro average F1 value [44] is used to compute the average F1 value across different label categories, which is more appropriate as an evaluation metric when the dataset samples are unbalanced. It gives the same weight to each different label and the formula for this evaluation metric is shown below:

$$F1 = \frac{1}{|C|}\sum_{t \in C}\frac{2P_t R_t}{P_t + R_t} \tag{21}$$

$$P_t = \frac{TP_t}{TP_t + FP_t} \tag{22}$$

$$R_t = \frac{TP_t}{TP_t + FN_t} \tag{23}$$

Here, TP_t, FP_t, FN_t represent the tth true positive, false positive, and false negative, respectively, in the label set C.

4.4. Baselines

To validate the effectiveness and advancements of the proposed model, we will compare it with the following 17 models in the next subsection. In this subsection, we will briefly introduce each of these 17 models. Notably, the last seven models utilize BERT for word embeddings.

LSTM [45]: The paper's authors implemented the LSTM model for target-related sentiment classification, utilizing the last hidden state vector for prediction. It has proven to be highly effective for aspect-based sentiment classification.

TD-LSTM [46]: This approach employs two LSTM networks to model the context surrounding the target: one in the forward direction and the other in the reverse direction. By connecting the final hidden states of these networks, the model can effectively predict the emotional polarity of the target.

ATAE-LSTM [47]: By integrating attention and embedding aspectual words individually as inputs, the model is able to efficiently perform the aspectual word sentiment categorization task.

MemNet [48]: The paper introducing the model presented a novel approach that utilizes location coding to represent location information. This coding is based on the model's distinctive multi-hop attention mechanism.

AOA [49]: By utilizing the attention-attention module [50], the model enables the interaction between the target representation and the textual representation generated by the LSTM. This interaction allows for the capturing of information regarding the interplay between the aspect and the contextual sentence, facilitating the joint learning of aspect and sentence representations.

IAN [51]: The attention mechanism within the model is responsible for computing the contextual representation by engaging with the target similarity. Consequently, the interaction information present in the context is employed to supervise the modeling of the target.

MCRF-SA [52]: By utilizing the CRF method, the model successfully extracts the opinion span pertaining to specific aspects of the text. Subsequently, the extracted opinion features and contextual information

Table 2
Dataset statistics.

Dataset	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	Test
Lap14	994	341	870	128	464	169
Rest14	2164	728	807	196	637	196
Rest15	1198	454	403	346	53	45
Rest16	1657	611	749	204	101	44
Twitter	1561	173	1560	173	3127	346

are employed to classify the sentiment polarity of the target.

ASGCN [53]: By leveraging Graph Convolutional Networks (GCNs) on the dependency tree of a sentence, the model effectively incorporates grammatical information. This approach allows for the utilization of long-distance word dependencies and the efficient retention of specific aspect features.

GL-GCN [54]: This model employs a Global and Local Dependency-Guided Graph Convolutional Network (GL-GCN) to integrate global and local dependency information. It captures local structural features using syntactic dependency structures and BiLSTM-generated sequence information, while also extracting global dependencies by constructing a word-document graph across the corpus. An attention mechanism is used to effectively combine these global and local signals.

CRF-GCN [33]: This model leverages a CRF chain to extract opinion spans related to specific aspect terms and utilizes a multi-layer GCN, enhanced by a refined position decay function, to integrate the contextual information from these spans into a global node. The sentiment polarity of each aspect is then predicted based on the vector representation of this global node. To handle accuracy fluctuations caused by multiple aspect terms in a sentence, the model introduces a global node within the GCN architecture.

BERT-BASE [13]: This model, built on the Transformer architecture, employs two innovative pre-training tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). These tasks enable BERT to acquire rich linguistic knowledge and contextual information. In Aspect-Based Sentiment Analysis (ABSA), BERT functions as a feature extractor, deriving word vector features from input text. Its pre-training enhances understanding of contextual relationships between words, thereby improving sentiment analysis accuracy.

RGAT-BERT [55]: Proposes a novel relationship graph attention network to encode sentence dependencies and make connections between aspects and opinions. Moreover, it proposes an aspect-oriented dependency tree structure. This experiment was rerun, using the parameters provided by the authors.

SPC-BERT [37]: This model uses a pretrained BERT classification model to generate word vectors and extracts the vectors corresponding to the sign bit [CLS] for final classification based on each vector output from the BERT model containing full text information.

AEN-BERT [37]: This model uses BERT for pretraining to generate context and target word vectors and an attention encoder to model the context and target.

LCF-BERT [37]: This model uses a local context-focusing mechanism that uses a dynamic masking layer for contextual features and a dynamic weighting layer for contextual features to focus on local contextual words.

UIKA [56]: This model addresses domain bias between pre-training and downstream tasks through a unified alignment pre-training framework. It introduces two key innovations: a coarse-to-fine retrieval sampling method for instance-level alignment, and a knowledge-guided strategy using a dual-model approach (knowledge-guiding and learner models) for knowledge-level alignment. An immediate teacher-student joint fine-tuning method facilitates knowledge transfer on the target dataset, enabling the learner model to retain domain-invariant knowledge while adapting to specific tasks.

MTABSA [22]: This model innovatively integrates Aspect Term Extraction (ATE) and Aspect Polarity Classification (APC) into a unified multi-task learning framework. It employs Multi-Head Attention (MHA) to incorporate dependency syntax information, enhancing aspect extraction by emphasizing crucial dependency relations. This approach enables simultaneous aspect term extraction and sentiment polarity classification while focusing on aspect-related words more effectively.

4.5. *Comparative experiment*

Table 3 presents the performance of 18 different models across 5 distinct datasets. We evaluated the models using the accuracy (ACC) and macro-average F1 (Macro-F1) metrics, as introduced in Section 4.3. Overall, our model demonstrates robust performance and reliability. Except for the accuracy on the Rest16 dataset, our model exhibits state-of-the-art results across all other datasets.

Generally, high scores are predominantly observed in models utilizing BERT for word embedding, while models employing alternative word embedding methods show less favorable results. Among the top-performing models are ASGCN, GL-GCN, and CRF-GCN, all of which are based on graph convolutional neural networks. These models demonstrate a significant advantage in accuracy on the Rest16 dataset compared to BERT-encoded models, suggesting that graph networks' dependency tree construction method offers notable benefits for aspect-based sentiment analysis.

A close examination of Table 3 clearly illustrates BERT's powerful feature representation capabilities. The consistently higher F1 scores demonstrate that BERT-encoded models generally achieve better balance compared to non-BERT-encoded models. Even BERT(base) alone shows impressive performance on several datasets. Our model, which incorporates CABiLSTM on top of BERT, achieves substantial performance improvements. Notably, our model achieves the highest F1 score on Rest14, the best accuracy on Rest15, the top F1 score on Rest16, and the highest accuracy on the Twitter dataset. These results validate the effectiveness and advanced nature of our proposed model.

Furthermore, our model utilizes a frozen BERT during training, which offers advantages in terms of resource consumption and convergence speed compared to models like MTABSA-BERT and UIKA-BERT that fine-tune BERT parameters during training. This approach provides a significant advantage in terms of training costs and efficiency.

In conclusion, our model not only demonstrates superior performance across various metrics and datasets but also offers practical benefits in terms of computational efficiency, solidifying its position as a state-of-the-art approach in aspect-based sentiment analysis.

4.6. *Ablation study*

To rigorously evaluate the efficacy of our proposed enhancements, we conducted a series of comprehensive ablation experiments on the CALSTM-BERT model. These experiments were designed to systematically assess the impact of each individual component on the overall performance of the model. Our primary focus was on examining the CABiLSTM architecture, with the aim of validating the effectiveness of our proposed improvements.

In Table 4, we present a detailed breakdown of our experimental results, with particular emphasis on the top two performing configurations. This presentation format enables a clear visualization of the relative importance of each model component and provides insights into the synergistic effects of different architectural choices.

We began by removing the entire CABiLSTM module, leaving only the BERT(base) model with its weights frozen, and then compared the performance of the CABiLSTM-BERT model against the BERT(base) model alone. As clearly demonstrated by the data in Table 4, the model's performance saw a significant improvement. While the output of BERT (base) consists of text sequence vectors enriched with contextual feature information, it still lacks the capability to effectively separate certain implicit features, which need to be mapped into higher-dimensional linear spaces for further extraction to enhance model performance. The CABiLSTM module, through its multi-layer, bi-directional LSTM architecture, facilitates the mapping of features into these higher-dimensional spaces, enabling the extraction of a wider range of implicit features. Additionally, the attention mechanism highlights the important features, and convolutional operations merge and retain the most useful features. This combination allows the model to capture valuable information that BERT(base) alone struggles to obtain, thereby significantly enhancing overall model performance.

The ABiLSTM-BERT model builds upon the BiLSTM-BERT(concat)

Table 3
Model comparison results.

Embedding	Model	Lap14		Rest14		Rest15		Rest16		Twitter	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
w/o BERT	LSTM	69.28	63.09	78.13	67.47	77.37	55.17	86.8	63.88	69.56	67.7
	TD-LSTM	71.8	68.46	78.00	68.43	76.39	58.7	82.16	54.21	/	/
	ATAE-LSTM	68.88	63.93	78.6	67.02	78.48	62.84	83.77	61.71	/	/
	MemNet	70.64	65.17	79.61	69.64	77.31	58.28	85.44	65.99	71.48	69.9
	AOA	72.62	67.52	79.97	70.42	78.17	57.02	87.50	66.21	72.3	70.2
	IAN	72.05	67.38	79.26	70.09	78.54	52.65	84.74	55.21	72.5	70.81
	MCRF-SA	74.01*	68.72*	80.62*	70.34*	78.68*	59.10*	87.28*	63.38*	/	/
	ASGCN-DG	75.55	71.05	80.77	72.02	79.89	61.89	**88.99**	67.48	72.15	70.40
	GL-GCN	76.91	72.76	82.11	73.46	80.81	64.99	**88.47**	69.64	73.26	71.26
	CRF-GCN	75.83	**74.78**	82.71	73.83	80.85	63.12	87.54	67.87	/	/
w BERT	BERT(base)	73.35*	66.38*	79.29*	66.51*	77.53	58.46	82.11*	56.43*	69.22*	67.20*
	RGAT-BERT	77.12*	72.11*	**84.02***	**75.49***	/	/	/	/	73.99*	**72.93***
	SPC-BERT	77.81	73.41	81.02	72.81	/	/	/	/	73.12	71.32
	AEN-BERT	77.19	72.57	81.69	73.64	/	/	/	/	73.7	72.10
	LCF-BERT	**78.12**	**73.43**	82.13	74.52	/	/	/	/	73.84	72.87
	UIKA-BERT	76.98*	72.55*	81.65*	75.21*	**82.76***	**67.54***	85.64*	**72.37***	73.98*	72.81*
	MTABSA-BERT	77.43*	72.07*	83.39*	75.09*	/	/	/	/	**74.13***	**73.07***
	Ours	**77.91**	73.04	**83.75**	**75.87**	**82.84**	**66.10**	86.61	**73.54**	**74.28**	72.20

Table 4
Ablation study.

Model	Lap14		Rest14		Rest15		Rest16		Twitter	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
BERT(base)	73.35	66.38	79.29	66.51	77.53	58.46	82.11	56.43	69.22	67.2
LSTM-BERT	73.79	66.88	79.75	69.33	77.99	58.49	83.98	61.37	70.45	68.39
BiLSTM-BERT	75.68	67.74	80.62	71.45	78.86	60.63	83.55	62.21	70.41	68.36
BiLSTM-BERT(concat)	75.27	67.15	80.66	71.1	79.52	61.96	84.01	67.13	71.36	70.01
ABiLSTM-BERT	75.42	68.02	81.23	**73.94**	**81.67**	62.87	**85.67**	70.52	72.89	71.48
CBiLSTM-BERT	**76.36**	**72.36**	**82.72**	73.25	80.8	**64.59**	84.48	**71.63**	**74.03**	**72.97**
CABiLSTM-BERT	**77.91**	**73.04**	**83.75**	**75.87**	**82.84**	**66.1**	**86.61**	**73.54**	**74.28**	**72.2**

architecture by calculating multi-head self-attention for each group of retained features. This enhances the visibility of key features within each group, facilitating subsequent classification tasks. In contrast, the CBiLSTM-BERT model, when compared to BiLSTM-BERT(concat), employs convolutional layers instead of the linear layer used for dimensionality reduction after feature concatenation. The convolution not only combines features but also provides the model with relative positional information that is absent when using linear layers. Consequently, both ABiLSTM-BERT and CBiLSTM-BERT exhibit a marked improvement over BiLSTM-BERT.

The CABiLSTM-BERT, which synergizes the strengths of ABiLSTM-BERT and CBiLSTM-BERT, exhibits high performance across the five datasets, thereby significantly underscoring the effectiveness of the proposed enhancements in this study.

We have highlighted the top 2 scores. The data with * in the table are the results we reproduced using the source code, which may differ from the results shown in the original paper due to the influence of various reasons such as our experimental equipment and random numbers.

5. Discussion

To delve deeper into the performance of the model and to understand the impact of key parameters and modules on the experimental outcomes, as well as to identify the limitations of the proposed model, this section provides a comprehensive discussion. Firstly, in Section 5.1, a case study was conducted to analyze the model's performance across various scenarios, including single-sentiment single-polarity, multi-sentiment single-polarity, and multi-sentiment multi-polarity cases, which were compared with two representative models on a per-instance basis. Subsequently, in Section 5.2, the influence of the number of LSTM layers on model performance was examined. In Section 5.3, the effect of the number of heads in the multi-head self-attention mechanism on the model's effectiveness was analyzed. Lastly, in Section 5.4, the limitations of the proposed model in terms of classification balance were discussed, with tests conducted to identify the underlying reasons.

5.1. Case study

To assess the limitations and performance of the current model, we conducted a series of case studies in this section to evaluate its effectiveness, comparing it with the widely cited classic model ASGCN and the well-performing MTABSA model. Table 5 lists the cases, aspects, ground truth labels, and model predictions. In the case studies, the target aspects were highlighted in red to stand out within the original sentences.

Case 1 represents a positive review, but it does not explicitly state a sentiment towards "ssd" at a close reading. However, semantically, the sentiment towards "ssd" is positive. This makes it challenging for ASGCN to capture the true sentiment polarity in the semantics. Our model and MTABSA, on the other hand, are able to accurately predict, indicating that both models have a certain degree of understanding of the semantics of the text.

In Case 2, all the adjectives are used to praise the "atmosphere," with less explicit sentiment towards other aspects, especially "perfect," which is too close to "drinks" and can easily cause interference. Consequently, ASGCN and MTABSA are visibly influenced by this interference, leading to incorrect predictions. Additionally, MTABSA also made an error in the case analysis of the original paper [22] due to this scenario. Our model, however, did not err in this instance. The reason may lie in the model focus on the "drinks" after multi-feature extraction and the highlighting of key points by the attention mechanism, which dilutes and separates the impact of the unrelated parts.

In Case 3, the first half of the text is negative, with only the final part about "pialla" being positive. This makes the model susceptible to the

Table 5
Case study.

No	Case	Aspect	Actual	Prediction		
				ASGCN	MTABSA	Ours
1	Performance is much better on the pro, especially if you install an ssd on it.	ssd	positive	neutralÍ	positiveP	positiveP
2	A beautiful atmosphere, perfect for drinks and/or appetizers.	drinks	neutral	positiveÍ	positiveÍ	neutralP
3	The food is just okay, and it's almost not worth going unless you are getting the pialla, which is the only dish that is really good.	pialla	positive	neutralÍ	positiveP	positiveP
4	There was a little difficulty doing the migration as the firewire cable system cannot be used with the ibook.	firewire cable system	negative	neutralÍ	negativeP	negativeP
		ibook	neutral	neutralP	negativeÍ	negativeÍ
5	Macbook notebooks quickly die out because of their short battery life, as well as the many unknowable background programs.	Macbook notebooks	negative	neutralÍ	negativeP	negativeP
		battery life	negative	negativeP	negativeP	negativeP
		background programs	negative	negativeP	negativeP	negativeP
6	One night I turned the freaking thing off after using it, the next day I turned it on, no GUI, screen all dark, power light steady, hard drive light steady and not flashing as it usually does.	GUI	negative	negativeP	negativeP	negativeP
		screen	negative	negativeP	negativeP	negativeP
		power light	neutral	positiveÍ	positiveÍ	negativeÍ
		hard drive light	negative	neutralÍ	negativeP	negativeP

negative sentiment of the first half, potentially leading to misclassification.

Case 4 exemplifies a multi-aspect, multi-polarity scenario. Here, "firewire cable system" is marked as negative, while "ibook" is neutral. The overall sentiment of the sentence is negative, with the emphasis on the negativity being directed towards the "firewire cable system," without a specific sentiment towards "ibook," thus categorizing "ibook" as neutral. The strong polarity towards a single aspect can significantly influence the model's interpretation of aspects that do not exhibit a clear sentiment.

Case 5 represents a multi-aspect single-polarity scenario, where while there are multiple sentiments, the overall tone of the sentence is overwhelmingly negative. This makes it relatively straightforward for the model to classify each sentiment as negative.

The final case presents a multi-aspect multi-polarity sentence structure. An ambiguously neutral aspect is sandwiched between three negative aspects, and without additional context, it is challenging to categorize "power light steady" as neutral. ASGCN and MTABSA interpreted the "power light steady" as positive, while our model appears to have been influenced by the preponderance of negative aspects, classifying it as negative.

Analysis of the cases and comparisons in Table 5 indicates that our model exhibits a certain advantage over the other two models. Particularly in single-aspect single-polarity texts where the noun phrase does not explicitly express an opinion, our model successfully retains the important features of the relevant aspects and accurately classifies the polarity, highlighting the importance of minimizing the loss of implicit characteristics. However, in the case of multi-aspect multi-polarity texts, our model, like the others, still encounters issues with misclassification, which may be addressed in our future research efforts.

5.2. The impact of the number of LSTM layers

To investigate the impact of the number of LSTM layers on the model's effectiveness, this section conducts experiments by adjusting the LSTM layer count. Due to the characteristics of the proposed CABiLSTM, changing the number of LSTM layers also affects the number of heads in the multi-head self-attention mechanism and the number of convolutional kernels. To ensure that the model is only affected by the number of LSTM layers in the experiments, all other parameters of the model are fixed, and the experiments are conducted only on the same dataset. The range of LSTM layer counts is adjusted from 1 to 4. The purpose of this experiment is to analyze the relationship between the number of LSTM layers and the overall performance of the model.

As depicted in Fig. 4, the CABiLSTM-BERT model with two layers of LSTM performs optimally across the five datasets. Theoretically, the more layers our proposed model has, the more implicit features it can extract, and the more model parameters there are. However, due to the constraints of data volume, an excessive number of layers and parameters can lead to insufficient training, resulting in underfitting and suboptimal performance. After conducting experiments and analyses, we conclude that, for the five datasets under consideration, the LSTM layer count of 2 layers in our model yields the best performance.

5.3. The impact of the number of heads

The performance of the proposed CABiLSTM-BERT model in this paper is influenced to some extent by the number of heads in its multi-head self-attention mechanism. To investigate this influence, we conducted experiments on the same dataset with all other model parameters fixed, evaluating the model's performance under different numbers of self-attention heads (including 2, 4, 8, and 12). The objective of this section is to analyze how the number of heads in the self-attention mechanism affects the overall performance of the CABiLSTM-BERT model.

As evident from Fig. 5, the model performs best on the five datasets when the number of heads in the attention mechanism is 8. The number of heads in the attention mechanism reflects, to some extent, the size of the feature space that the model maps to. A larger feature space implies a stronger capability of the model to represent complex information, along with an increase in the number of model parameters. Similar to the situation in Section 5.2, due to the limitations of data volume, our model cannot use an excessive number of parameters, as this could lead to overfitting. Consequently, through experiments, we conclude that, for the five datasets considered, the optimal number of heads for our proposed model is 8.

5.4. Analysis of the limitations of balanced classification

In the experiments conducted in this paper, we employed the Lap14, Rest14, Rest15, Rest16, and Twitter datasets, each of which contained only data from the three categories of positive, negative, and neutral. The distribution of data quantities is depicted in Fig. 7. It is evident from Fig. 7 that the sample counts for each category in all datasets are imbalanced. Among these, the Twitter dataset has relatively better category balance, resulting in the accuracy and F1 scores calculated for almost all models on the Twitter dataset being the closest. In contrast, the SemEval series datasets have poor balance, as seen in Fig. 7, where the most numerous category is Positive, and the sample count for the Neutral category is always the least, particularly in Rest15 and Rest16,

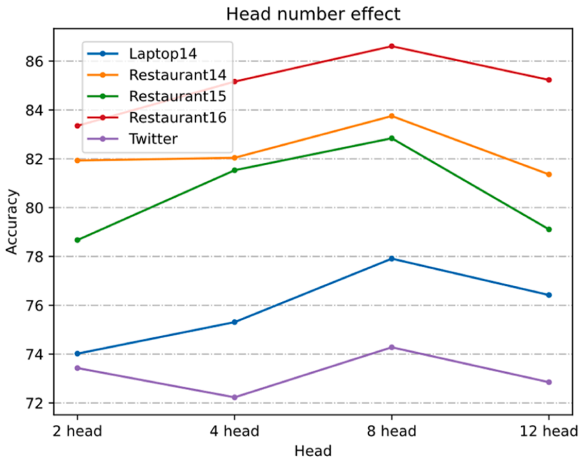

Fig. 5. The impact of the number of heads.

where the Neutral category samples are so sparse as to be almost negligible. Additionally, the proportion of Negative category samples in all Rest datasets is also low, which significantly impacts the balance of model classification.

To investigate the impact of the imbalanced datasets on the classification effectiveness of our proposed model, we used the model to predict the classification of each category within each dataset individually, resulting in the model's classification accuracy for each category on the four datasets, as depicted in Fig. 6. A careful examination of Figs. 6 and 7 reveals that the imbalance in the dataset data significantly affects the classification balance of our model. It is noteworthy that the category with the most samples in the training set consistently achieves the highest accuracy, and the classification accuracy for each class is proportional to the size of its data. For instance, the Rest15 dataset contains only a very small number of Neutral class samples, leading to extremely low classification accuracy for that class in our model. Therefore, these findings underscore the significant impact of dataset imbalance on our model's classification balance. It is worth mentioning that this issue is not unique to our model; many current models also face similar challenges. Addressing this issue will be one of our future research priorities.

6. Conclusion

This paper focuses on the aspect-based sentiment analysis task and driven by the approach of fully leveraging implicit feature information at each layer to minimize information loss and enhance model performance, we introduce a novel model named CABiLSTM-BERT. This advanced model is specifically designed to address these fine-grained sentiment analysis tasks. To enhance the capture of implicit feature representations within the input sequence and minimize information loss, CABiLSTM employs multiple LSTM layers to separately extract and retain a wealth of implicit feature information. Moreover, the latter part of the CABiLSTM utilizes multi-head self-attention and convolutional calculations to highlight salient features and effectively integrate them, thereby reducing information loss. Extensive experiments conducted on five publicly available real-world datasets yielded encouraging results, demonstrating the superior performance of the proposed method.

For future research directions, firstly, while our model outperforms other models in multi-aspect, multi-level sentiment classification, it is not yet optimal. This could be due to the interplay of sentiments across different aspects and levels during model computation. Future research should aim to minimize the impact of such interplay, which could significantly enhance the model's performance in this domain.

Secondly, the classification balance of our model is greatly influenced by the data balance of the dataset, which is a common issue for most current models. The prevalent approach to address this is to adjust the dataset's data distribution, starting from the data itself. However, this method is inherently limited and struggles to address the root cause of the problem. Consequently, designing models to mitigate the impact of dataset data imbalance on model classification balance will also be a focus of our future work.

CRediT authorship contribution statement

Bo He: Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Investigation, Funding acquisition, Formal analysis. **Ruoyu Zhao:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation. **Dali Tang:** Writing – review & editing, Validation, Methodology, Investigation, Formal analysis.

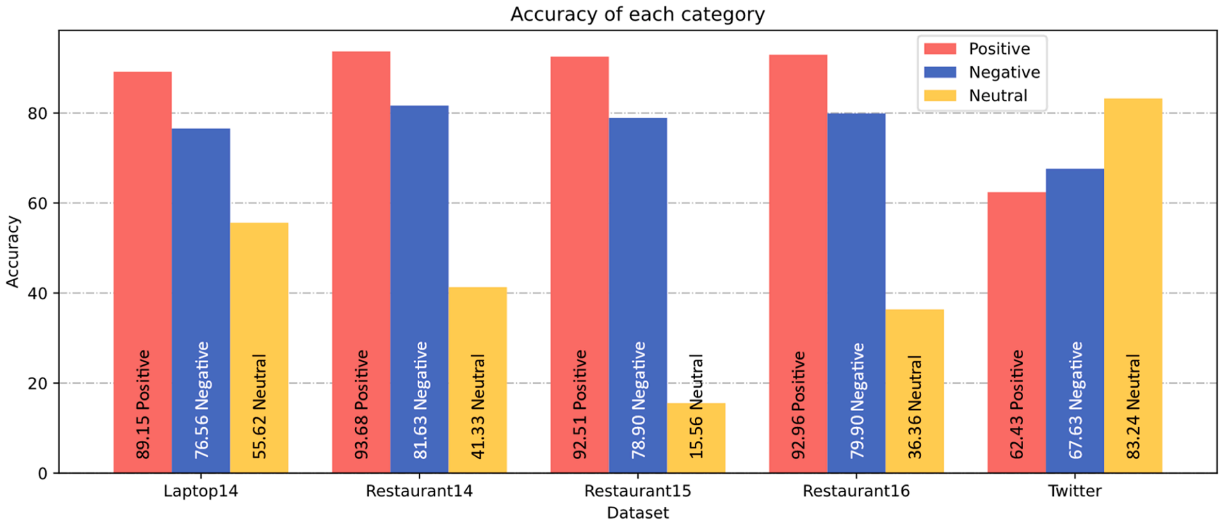

Fig. 6. Accuracy for each category.

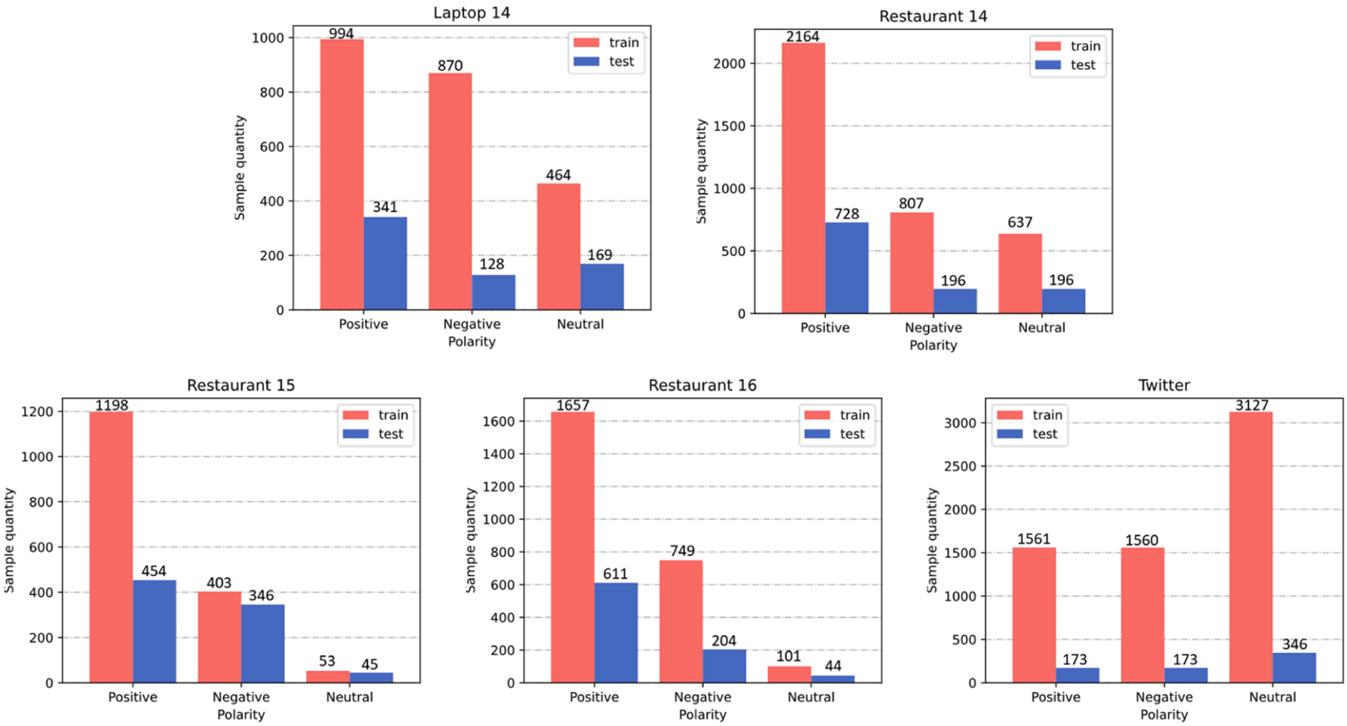

Fig. 7. Number of examples for each category.

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.

Acknowledgements

This work was supported by the 2023 High-Quality Innovation Project for Postgraduates of Chongqing University of Technology (gzlcx20233249).

Data availability

Data will be made available on request.

References

[1] Y. Zhang, T. Li, Review of comment-oriented aspect-based sentiment analysis, Comput. Sci 47 (7) (2020).

[2] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, Sentiment and sarcasm classification with multitask learning, in: IEEE Intelligent Systems 34, 2019, pp. 38–43, https://doi.org/10.1109/MIS.2019.2904691, 1 May-June.

[3] T. Jiang, Z. Wang, M. Yang, et al., Aspect-based sentiment analysis with dependency relation weighted graph attention, Information 14 (3) (2023) 185.

[4] A. Nazir, Y. Rao, L. Wu, L. Sun, Issues and challenges of aspect-based sentiment analysis: a comprehensive survey, IEEE Trans. Affect. Comput. 13 (2) (2022) 845–863. Second Quarter.

[5] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, B. Gupta, Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews, J. Comput. Sci. 27 (2018) 386–393.

[6] T. Brychcín, M. Konkol, J. Steinberger, Uwb: machine learning approach to aspect-based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval, 2014, pp. 817–822, 2014.

[7] W. Wang, N. Yang, F. Wei, B. Chang, M. Zhou, Gated self-matching networks for reading comprehension and question answering, in: Pro-ceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 189–198.

[8] G. Brauwers, F. Frasincar, A survey on aspect-based sentiment classification, ACM Comput. Surv. 55 (4) (2022) 1–37.

[9] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, B. Gupta, Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews, J. Comput. Sci. 27 (2018) 386–393.

[10] R.K. Yadav, L. Jiao, M. Goodwin, O.-C. Granmo, Positionless aspect based sentiment analysis using attention mechanism, Know-Based Syst. 226 (2021) 107136.

[11] Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L. (2018) Deep con-textualized word representations. CoRR, abs/1802.05365.

[12] Alec R., Karthik N., Tim S., Ilya S. (2018) Improving language understanding by generative pre-training. The university of british columbia vancouver campus, Vancouver.

[13] Devlin J., Chang M.W., Lee K., et al. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[14] W. Zhang, X. Li, Y. Deng, et al., A survey on aspect-based sentiment analysis: tasks, methods, and challenges, IEEE Trans. Knowl. Data Eng. 35 (11) (2022) 11019–11038.

[15] A. Karimi, L. Rossi, A. Prati, Adversarial training for aspect-based sentiment analysis with bert, in: 2020 25th International conference on pattern recognition (ICPR), IEEE, 2021, pp. 8797–8803.

[16] D. Anand, D. Naorem, Semi-supervised aspect based sentiment analysis for movies using review filtering, Procedia Comput. Sci. 84 (2016) 86–93.

[17] S. Kiritchenko, X. Zhu, C. Cherry, S.M. Mohammad, Detecting aspects and sentiment in cus-tomer reviews, in: 8th International Workshop on Semantic Evaluation (SemEval), 2014, pp. 437–442.

[18] S. Poria, N. Ofek, A. Gelbukh, A. Hussain, L. Rokach, Dependency tree-based rules for concept-level aspect-based sentiment analysis. Semantic Web Evaluation Challenge, Springer, Cham, 2014, pp. 41–47.

[19] A. Weichselbraun, S. Gindl, A. Scharl, Extracting and grounding contextualized sentiment lexi-cons, IEEE Intell. Syst. 28 (2) (2013) 39–46.

[20] L. Zheng, H. Wang, S. Gao, Sentimental feature selection for sentiment analysis of Chinese online reviews, Int. J. Mach. Learn. Cybern. 9 (1) (2018) 75–84.

[21] J. Liu, S. Zheng, G.X. Xu, M.W. Lin, Cross-domain sentiment aware word embeddings for review sentiment analysis, Int. J. Machine Learn. Cybernet. 12 (2) (2021) 343–354.

[22] G. Zhao, Y. Luo, Q. Chen, X. Qian, Aspect-based sentiment analysis via multitask learning for online reviews, Knowl Based Syst. 264 (2023) 110326.

[23] J. Zhou, J.X. Huang, Q. Chen, H. Qinmin Vivian, T. Wang, L. He, Deep learning for aspect-level sentiment classification: survey, vision, and challenges, IEEE Access 7 (2019) 78454–78483.

[24] N. Li, C.Y. Chow, J.D. Zhang, SEML: a semi-supervised multi-task learning framework for aspect-based sentiment analysis, IEEE Access 8 (2020) 189287–189297.

[25] Z. Chen, T. Qian, Transfer capsule network for aspect level sentiment classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 547–556.

[26] S. Jindian, Shanshan Yu, D. Luo, Enhancing aspect-based sentiment analysis with capsule network, IEEE Access 8 (2020) 100551–100561.

[27] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 2: Short papers), 2014, pp. 49–54.

[28] Vaswani A., Shazeer N., Parmar N.,et al. Attention Is All You Need. arXiv, 2017.
[29] F. Zhao, Z. Wu, X. Dai, Attention transfer network for aspect-level sentiment classification, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 811–821.
[30] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, Aspect-level sentiment analysis via convolution over dependency tree, 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. 5679–5688.
[31] B. Liang, H. Su, L. Gui, E. Cambria, R. Xu, Aspect-based sentiment anal-ysis via affective knowledge enhanced graph convolutional networks, Knowl.-Based Syst 235 (2022) 107643.
[32] Z. Zhang, Z. Zhou, Y. Wang, Ssegcn: syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022, pp. 4916–4925.
[33] B. Huang, J. Zhang, J. Ju, R. Guo, H. Fujita, J. Liu, CRF-GCN: an effective syntactic dependency model for aspect-level sentiment analysis, Knowl Based Syst 260 (2023) 110125.
[34] J. Zeng, T. Liu, W. Jia, J. Zhou, Relation construction for aspect-level sentiment classification, Inform. Sci. 586 (2022) 209–223.
[35] H.T. Phan, N.T. Nguyen, D. Hwang, Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis, Inform. Sci. 589 (2022) 416–439.
[36] M. Zhao, J. Yang, J. Zhang, S. Wang, Aggregated graph convolutional networks for aspect-based sentiment classification, Inform. Sci. 600 (2022) 73–93.
[37] S. Feng, B. Wang, Z. Yang, J. Ouyang, Aspect-based sentiment analysis with attention-assisted graph and variational sentence representation, Knowl.-Based Syst (2022) 109975.
[38] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, 15 Nov.
[39] Pontiki, M., Galanis, D., Pavlopoulos, J., et al. (2014). Semeval-2014 task 4: aspect based sentiment analysis. In SemEval 2014 (pp. 27–35).
[40] Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., & Androutsopoulos, I. (2015). Semeval-2015 task 12: aspect based sentiment analysis. In SemEval 2015 (pp. 486–495).
[41] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, et al., Semeval-2016 task 5: aspect based sentiment analysis, In SemEval (2016) 19–30.
[42] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent Twitter sentiment classification, in: Proceedings of the 52nd annual meeting of the association for computational linguistics, 2014, pp. 49–54.
[43] Loshchilov I., Hutter F. Fixing weight decay regularization in adam. arxiv preprint arxiv:1711.05101, 2017, 5.
[44] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, et al., Large-scale hierarchical text classification with recursively regularized deep graph-CNN, in: Proceedings of the 27th international conference on world wide web, 2018, pp. 1063–1072.
[45] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, in: Proceedings of the 25th international conference on computational linguistics, 2016, pp. 3298–3307.
[46] D. Tang, B. Qin, X. Feng, and T. Liu, ‘‘Effective LSTMs for target-dependent sentiment classification,’’ 2015, arXiv:1512.01100. [Online].
[47] 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.
[48] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 214–224.
[49] Huang, B., Ou, Y., & Carley, K.M. (2018). Aspect level sentiment classification with attention-over-attention neural networks. In SBP-BRiMS 2018 (pp. 197–206).
[50] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, G. Hu, Attention-over-attention neural networks for reading comprehension, in: Proceedings of the 55th annual meeting of the association for computational linguistics, 2017, pp. 593–602.
[51] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the 26th international joint conference on artificial intelligence, 2017, pp. 4068–4074.
[52] L. Xu, L. Bing, W. Lu, F. Huang, Aspect sentiment classification with aspect-specific opinion spans, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 3561–3567.
[53] C. Zhang, Q. Li, D. Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, in: Proceedings of the 2019 conference on empirical methods in natural language processing, 2019, pp. 4567–4577.
[54] X. Zhu, L. Zhu, J. Guo, et al., GL-GCN: global and local dependency guided graph convolutional networks for aspect-based sentiment classification, Expert Syst. Appl. 186 (2021) 115712.
[55] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3229–3238.
[56] J. Liu, Q. Zhong, L. Ding, et al., Unified instance and knowledge alignment pretraining for aspect-based sentiment analysis, IEEE/ACM Trans. Audio Speech Lang Process 31 (2023) 2629–2642.