The Journal of Supercomputing (2022) 78:14846–14865
https://doi.org/10.1007/s11227-022-04480-w

Graph convolutional networks with hierarchical multi-head attention for aspect-level sentiment classification

Xiaowen Li[1] · Ran Lu[1] · Peiyu Liu[1] · Zhenfang Zhu[2]

Accepted: 2 March 2022 / Published online: 9 April 2022

Abstract

Aspect-level sentiment classification has been widely used by researchers as a fine-grained sentiment classification task to predict the sentiment polarity of specific aspect words in a given sentence. Previous studies have shown relatively good experimental results using graph convolutional networks, so more and more approaches are beginning to exploit sentence structure information for this task. However, these methods do not link aspect word and context well. To address this problem, we propose a method that utilizes a hierarchical multi-head attention mechanism and a graph convolutional network (MHAGCN). It fully considers syntactic dependencies and combines semantic information to achieve interaction between aspect words and context. To fully validate the effectiveness of the method proposed in this paper, we conduct extensive experiments on three benchmark datasets, which, according to the experimental results, show that the method outperforms current methods.

Keywords Aspect-level sentiment classification · Deep learning · Graph convolutional network · Attention mechanism

✉ Ran Lu
luran@sdnu.edu.cn

Xiaowen Li
m17806098737@163.com

Peiyu Liu
liupy@sdnu.edu.cn

Zhenfang Zhu
zhuzf@sdjtu.edu.cn

[1] School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

[2] School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan 250357, China

1 Introduction

Sentiment analysis, a hot topic in the field of natural language processing, has generated a lot of interest. Due to the recent impact of the COVID-19 pandemic, people focus on social distance and use online commenting platforms more frequently, such as e-commerce platforms and micro-blogs. We can use such online commenting platforms to provide suggestions for comments about the type of sentiment content [1]. In the era of big data, a large amount of information data can be generated every second, especially textual big data. When processing these information data, data mining algorithms of deep learning are used to perform sentiment analysis on textual data. Most sentiment classification tasks are document-level and sentence-level, but a word may express opposite sentiment in different contexts, so aspect-level sentiment classification [2, 3] is considered to solve this kind of problem. Aspect-level sentiment classification (ALSC) is a fine-grained sentiment classification task that aims to identify the sentiment polarity (e.g. positive, negative, neutral) of a particular aspect of a sentence, and an example is shown in Fig. 1. For instance, in the sentence "The price of this location is very expensive, but the transportation is convenient", we can see that the affective polarity for the aspect "price" is negative, but for the aspect "transportation" is positive. However, the aspect of "price" does not need to be "convenient", and it even brings noise to the sentiment analysis of "price". Therefore, the task of ALSC is to find adjectives related to aspects so that the sentiment polarity of a given aspect can be predicted [4].

With the popularity of deep learning and the improvement of computer hardware devices nowadays, labeled data are gradually becoming huge, and deep learning models [5] have replaced many classical techniques for solving natural language processing. Deep learning models based on deep learning have achieved state-of-the-art performance in a variety of tasks, including sentiment analysis, machine translation and named entity recognition, as well as classification, image generation, image segmentation and unsupervised in image computer vision supervised feature learning, among others. In recent years, an increasing number of deep learning approaches have been explored for ALSC tasks that offer better scalability than traditional feature-based approaches [6, 7]. Recurrent neural network (RNN) [8] employs semantic combination functions, which enable them to handle the complex combinatoriality in sentiment analysis. Recurrent neural networks model the sequence information of sentences, obtain distant dependencies and generate representations of sentences to improve the accuracy of prediction by learning about the sequence. However, the problem of gradient disappearance in RNN network structure cannot be solved yet, and a better way to solve this problem is to use networks with Long Short-Term Memory (LSTM) [9, 10] or Gated Recurrent Unit (GRU) [11] architecture. LSTM is a special recurrent

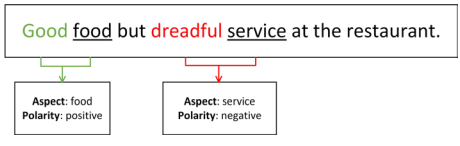

Fig. 1 An example of ALSC

neural network that can learn long-term dependency information, but LSTM still lacks sensitivity for some words and has no outstanding performance in sentiment analysis tasks. Compared with RNN, convolutional neural networks (CNN) [12] can capture local features of sentences and extract aspect-related information, but cannot establish deep semantic relationships between aspects and contexts.

After that, models based on attention mechanism also started to be applied in such tasks, which emphasize more on the importance of the model for the given aspect words. Through focusing on the opinion words that express the sentiment polarity of the aspect words in the sentence and reducing the attention to other non-opinion words, the model can avoid the influence of irrelevant noise information and make correct predictions of the sentiment polarity of the aspect words. However, the attention mechanism in the sentences is flawed and vulnerable to the noise generated by the attention mechanism. In addition, the attention mechanism cannot capture the syntactic dependencies between contextual words and aspects in a sentence, because some irrelevant words may receive more attention because of syntactic problems, and thus, some valuable and important information will be lost. It is important for the sentiment analysis task to model more effectively the semantic dependencies between aspect words and context words in sentences.

Although the combination of neural networks and attention mechanisms is of great significance in aspect-level sentiment classification, syntactic dependency relations between aspect words and context words are not available. Dependency trees can capture the long distance between aspect words and opinion words, better link the relationship between target words and sentiment words, and establish word-to-word connections, thus providing a differentiated syntactic path for information propagation on the tree, such as the existence of dependency relations between "price" and "reasonable". In recent years, scholars have become increasingly interested in the extension of deep learning methods to graphs, and researchers have borrowed ideas from convolutional neural networks, recurrent neural networks, and deep autoencoder to design a neural network structure for processing graph data—Graph Neural Networks (GNN). Graph Neural Networks flourish and are generally used in node classification and graph representation learning, such as Graph Convolutional Network (GCN) [13, 14], which successfully learns the representation of nodes, captures the local position of nodes in the graph, and views the dependency tree as an adjacency matrix. GCN is a convolutional neural network that operates on graphs and can capture interdependent information from the rich relational data. Graph attention network (GAT) introduces an attention mechanism in GNN to classify nodes of graph structure data and compute the hidden representation of each node by paying attention to its neighboring nodes. In GAT, different levels have different attention weights. Dependency tree-based graph convolutional networks and graph attention networks explicitly exploit the syntactic structure of sentences, and the dependency syntactic tree is equivalent to the structure of a graph, thus developing the current neural network. To exploit the syntactic information between aspect and contextual words, Zhang et al. [15] proposed a new aspect-specific sentiment classification framework by building a graph convolutional network on the

dependency tree of sentences, which incorporates dependency trees into the attention model to exploit syntactic information and word dependencies.

Despite the good experimental results of the previous study, there are still shortcomings that need to be improved. The attention mechanism may assign higher attention weight to words with strong emotional color, resulting in keywords with lower attention weight, so some important words may be ignored, so the noise problem will exist in the model and affect the judgment of sentiment polarity. In this paper, we introduce a hierarchical attention mechanism to avoid the loss of important information. In addition, these methods ignore the syntactic relationships between aspects and the corresponding contextual words, leading the model to incorrectly focus on syntactically irrelevant words. The GCN contains useful information for identifying syntactic relationships, but it assigns the same weight to all edges between connected words. By iterating on graph convolution propagation, it may incorrectly associate target aspects with irrelevant words.

Our main contributions are summarized as follows:

1. We propose a graph convolutional network based on dependency trees, which makes full use of syntactic information and effectively captures the syntactic dependencies between aspect words and contexts.
2. We propose a hierarchical multi-headed attention mechanism that fully considers the semantic relationships between aspect words and contexts, and excludes the influence of contextual words that are not related to aspect words.
3. We conducted extensive experiments on three benchmark datasets to validate the model MHAGCN and analyze the advantages of the model over other state-of-the-art methods.

The rest of our paper will be organized by the following rules: In Sect. 2, we describe the related work about aspect-level sentiment classification in detailed. Section 3 introduces the model method we proposed. Experiment detail and results are discussed in Sect. 4. Section 5 summarizes our work in the paper.

2 Related work

Aspect-level sentiment classification is a fine-grained sentiment classification that aims to predict the sentiment polarity of specific aspect words in a sentence. The traditional treatment is to build feature engineering for the model and select a good set of features. In early studies, traditional methods such as sentiment dictionaries and machine learning were generally used. Akhtar et al. [16] combined the output of multi-layer perceptron networks from deep learning and feature-based models to propose a stacked integration approach for predicting sentiment and emotion intensity. Support vector machine (SVM) [17] is a traditional machine learning method used to solve aspect sentiment classification with good results.

In recent years, deep learning models have received increasing attention because this generates dense vectors of sentences without manually constructing features,

automatically capturing important sentiment features from the text. Recently, deep learning model has been widely apply in aspect-level sentiment classification because of their obvious advantages in automatically learning text features, and it can avoid relying on manual design features and map features into continuous low-dimensional vectors in automatically learning text features. Xue et al. [18] proposed a convolutional neural network model based on gated mechanism, which can selectively export sentiment features on the basis of a given aspect or entity. Ruder et al. [19] pointed out that providing contextual information between different sentences can help the model better determine the comment text in multiple aspects of sentiment tendency. They proposed a comment hierarchical model based on aspect-level sentiment classification. This model exploited a hierarchical LSTM network for sentiment classification, which makes better use of the grammar features and aspects of the position information of the sentence. Zhang et al. [20] put forward two gated neural networks, one for capturing the syntactic and semantic information of tweet-level, and the other for modeling the interaction between the upper left and upper right context words of a given target, which is represented by the sentiment features of bidirectional GRU learning.

The attention mechanism uses the semantic relationship between aspect and context to calculate the attention weights of contextual words. Wang et al. [21] present AE-LSTM neural network and ATAE-LSTM neural network models to obtain contextual feature information through LSTM. ATAE-LSTM model based on attention and aspect word vector takes the aspect word vector as the attention target. The aspect feature representation is connected with the hidden state matrix after the sentence is modeled by LSTM. The attention weight of each time step is calculated by using the feed-forward hiding layer and constructs aspect-related expression of sentiment characteristics. Tang et al. [22] designed a deep memory network, in which target information is integrated by multiple computing layers. Each layer is an attention model based on context and location. Ma et al. [23] used two attention networks to model the mutual effect of aspects and contexts, which enhanced the interactive learning process of aspects and contexts. Fan et al. [24] proposed a fine granularity attention mechanism that captures the word-level interaction between the aspects and the context. Chen et al. [25] proposed a recurrent attention memory model (RAM). According to the distance information between the words and aspect words in the sentence, different position weights are assigned to the memory fragments produced by each word, finally using GRU network and multi-layer attention mechanism to structure in terms of sentiment characteristics of the representation.

Graph neural networks have a flexible structure and update that can represent some structural properties of the data itself well. GNNs are now also used in text summarization, text classification and sequence labeling tasks. GNNs and their variants have achieved good results on natural language processing tasks to better represent information in the model. Common graph neural network algorithms are mainly networks such as GCN and GAT and their variants. GCNs have proven to be effective models for many natural language processing applications such as relation extraction, reading comprehension and aspect-level sentiment analysis. Cai et al. proposed a hierarchical graph convolution model, including low-level GCN and high-level GCN, which are used to model the relationship between multiple

Table 1 Collection of related works

References	Year	Datasets	Adopted scheme
Akhtar et al. [16]	2020	SemEval-2017	MLP, deep learning
Kiritchenko et al. [17]	2014	Yelp restaurant and the Amazon laptop reviews	SVM
Xue et al. [18]	2018	SemEval-2014	Gated Convolutional Networks, CNN
Ruder et al. [19]	2016	SemEval-2016	LSTM
Zhang et al. [20]	2016	Twitter, MPQA	Gated Convolutional Networks
Wang et al. [21]	2016	SemEval-2014	Attention, LSTM
Tang et al. [22]	2016	Twitter	LSTM
Ma et al. [23]	2017	SemEval-2014	LSTM, Attention
Fan et al. [24]	2018	SemEval-2014 and Twitter	Attention
Chen et al. [25]	2017	SemEval-2014 and Twitter	RNN
Cai et al. [26]	2020	SemEval-2015, 2016	Graph convolution network
Zhang et al. [27]	2020	SemEval-2014, 2015, 2016 and Twitter	Graph convolution network

categories and capture the relationship between sentiment and aspect categories respectively. Zhang et al. combined hierarchical syntactic graphs and lexical graphs to capture global word co-occurrence information using lexical graphs and built conceptual hierarchies on both graphs to distinguish different types of dependencies. Dong et al. [26] proposed an architecture to propagate word-to-aspect word sentiment based on contextual words and syntactic structure. Phan et al. [27] proposed syntactic relative distance to mitigate the adverse effects of disjoint words for the adverse effect of sentiment prediction. Based on these ideas, researchers have extended graph neural network models based on syntactic dependency trees, and some excellent work has emerged (Table 1).

3 Model

3.1 Task definition

Aspect-level sentiment classification is to predict the sentiment polarity of the aspect word in a sentence based on contextual information. We are given a contextual sentence $S = \{w_1, w_2, \ldots, w_{n-1}, w_n\}$ with the aspect word $a = \{w_i, w_{i+1}, \ldots, w_{i+m-1}\}$. The aspect word can be either a word or a phrase.

3.2 Embedding

From Fig. 2, we use two pre-trained models to initialize the feature vector of each word. One is GloVe, which has been widely used in many neural network-based models for NLP tasks. The other is Bidirectional Encoder Representations from

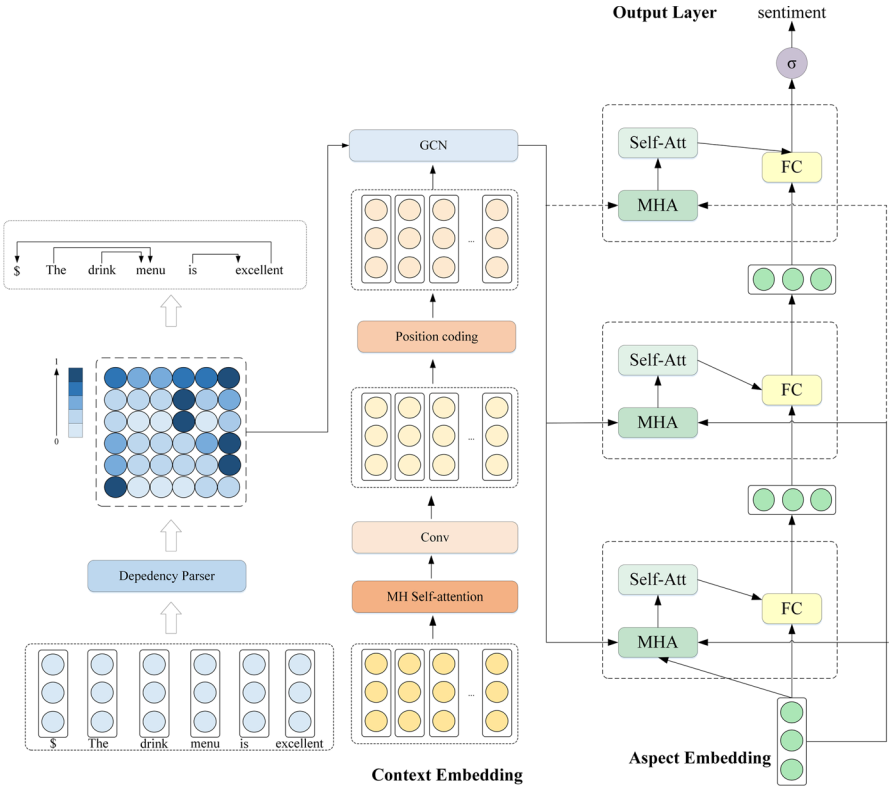

Fig. 2 The proposed MHAGCN model

Transformers (BERT), which is a pre-trained bidirectional transformer encoder with the advantage of sequence-to-sequence that has achieved state-of-the-art performance in various NLP tasks.

3.2.1 GloVe embedding

We use a pre-trained embedding matrix, Glove [28], to obtain a fixed word embedding for each word. We first map each input word w_i into a low-dimensional word embedding vector $e_i \in \mathbb{R}^{d_w}$. d_w is the dimension of the word vector and $l \in \mathbb{R}^{d_w \times |V|}$ is the pre-trained GloVe embedding matrix, where $|V|$ is the size of the vocabulary table.

3.2.2 BERT embedding

To facilitate the training and fine-tuning of the BERT [29] model, we reconstruct the given context and target as "[CLS] +context+ [SEP]" and "[CLS] +aspect+ [SEP]", which are fed into BERT. [CLS] is a token inserted at the beginning of a sentence and

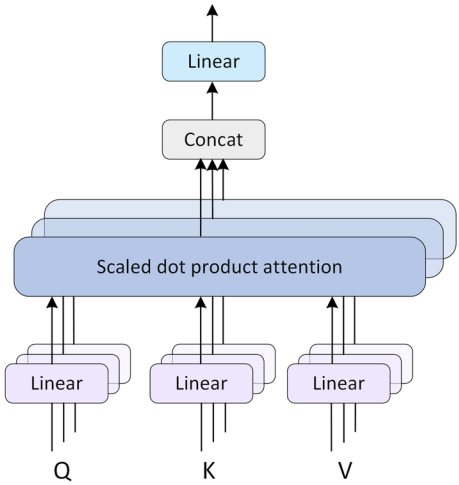

Fig. 3 The architecture of MHSA mechanism

[SEP] is a clause token used to separate two input sentences. Then, we use the average pooling method to aggregate the information carried by the words from BERT to obtain the final embedded words $X \in \mathbb{R}^{n \times d_B}$, d_B representing the dimension of the BERT output.

3.3 Multi-head self-attention

The self-attention mechanism mainly emphasizes on correlation, but how to further capture the correlation among the vectors, we then use a multi-headed attention mechanism. Instead of computing attention just once, multi-head self-attention passes scaled dot-product attention through multiple times in parallel, the outputs of the independent attention computing units are then simply stitched together and finally converted into appropriately sized dimensions by a linear unit. The architecture of MHSA is shown in Fig. 3.

We provide the query sequence $q = \{q_1, q_2, \ldots, q_n\}$ and key sequence $k = \{k_1, k_2, \ldots, k_n\}$. The attention function maps the key sequence k and the query sequence q to the output sequence:

$$Attention(k, q) = softmax(f(k, q)) \tag{1}$$

where $W_{att} \in \mathbb{R}^{d_h \times d_h}$ are learnable weights.

As shown in Fig. 3, Q, K and V first goes into the linear transformation and then is input to the scaled dot product attention. We perform this operation h times, which is the multi-head.

$$H_i = Attention(k, q) \tag{2}$$

$$MHSA = \left(H_1 \oplus H_2 \oplus \ldots \oplus H_h\right) \cdot W_h \tag{3}$$

$W_h \in \mathbb{R}^{d_h \times d_h}$ are the parameter matrices.

With the above analysis of MHSA, given the contextual embedding m_i^c, we can acquire the contextual representation processed by the attention mechanism:

$$c_i^s = MHSA\left(m_i^c, m_i^c\right) \tag{4}$$

The full contextual representation is as follows:

$$c^s = \left\{m_1^c, m_2^c, \ldots m_n^c\right\} \tag{5}$$

3.4 Convolution layer

The convolution layer can transform the contextual information collected by MHSA. Its double-layer structure is tightly coupled, with the activation function of the first layer being *Relu* and the activation function of the second layer being linear. For further analysis of context and aspect information, we transform them. The convolution operation is as follows:

$$Conv(m) = Relu\left(m * W^1 + b^1\right) * W^2 + b^2 \tag{6}$$

W^1 and W^2 are the learnable weight, b^1 and b^2 are the biases. [*] is the convolution operator.

We convert the output c^s of MHSA to h^c by convolution conversion section as follows:

$$h_i^c = Conv(c^s) \tag{7}$$

$$h^c = \left\{h_1^c, h_2^c, \ldots, h_n^c\right\} \tag{8}$$

3.5 Position coding

In general, the closer a word is to an aspect word, the more likely it is to be an opinion word, that is, the more likely it is to carry the sentiment information of the aspect word. Therefore, positional coding is introduced in the model to model the effect of position information on the prediction results.

$$weight_i = \begin{cases} 1, & dis = 0 \\ 1 - \frac{dis}{N}, & 1 \le dis \le s \\ 0, & dis > s \end{cases} \tag{9}$$

$$H(h^c) = weight_i h^c \tag{10}$$

3.6 Graph convolutional network

We use a graph convolutional network based on syntactic dependency tree so that efficient graph convolution can be used to encode the dependent syntactic structure of the input sentences. The graph convolution in the sentence dependency tree gives syntactic constraints to an aspect of the sentence to discriminate descriptive words based on syntactic distance. When the node representation passes through the GCN layer, the representation of each node is further enriched by the syntactic information of the dependency tree.

We construct a syntactic dependency tree using the spaCy toolkit[1] and then use the dependency tree transformation to obtain its corresponding adjacency matrix $M \in \mathbb{R}^{k \times k}$, k denotes the length of the sentence. In the L-layer GCNs, the input of the node i in the l-th layer is represented as follows:

$$h_i^l = ReLU\left(\sum_{j=1}^{k} M_{ij} W^l h_j^{l-1} + b^l\right) \tag{11}$$

For the L-layer GCNs, $l \in [1, 2, \ldots, L]$ and h_i^L the final state of node i. $h_j^{l-1}$is the representation of the j-th token evolved from the $(l-1)$-th GCN layer. The weight W^l are parameters that need to be learned and b^l is bias vector. We update the representation of each node by using a graph convolution operation with a normalization factor.

$$\tilde{h}_i^l = \sum_{j=1}^{n} M_{ij} W^l g_j^{l-1} \tag{12}$$

$$h_i^l = ReLU\left(\tilde{h}_i^l / \left(d_i + 1\right) + b^l\right) \tag{13}$$

$$d_i = \sum_{j=1}^{n} M_{ij} \tag{14}$$

$$g_i^l = H(h^c) \tag{15}$$

$g_j^{l-1} \in \mathbb{R}^{d_h \times d_h}$ is the representation of the j-th token and d_i is degree of the i-th token in the tree.

3.7 Hierarchical multi-head attention layer

This hierarchical multi-head attention allows combining aspect embedding with the input of the current attention layer, allowing the model to focus on the interaction

[1] https://spacy.io/.

between aspects and keywords in the context to prevent the effects of noise while preserving the aspect information. The hierarchical multi-head attention layer consists of multiple attention layers. Each attention layer has three modules for multi-headed attention, self-attention, and feature fusion, respectively. The input of each attention layer is the output of the previous layer, the output of the graph convolution network and the output of the aspect embedding.

3.7.1 Multi-head attention

Multi-head attention (MHA) allows the model to jointly focus on different information from different locations. It captures the semantic information of the context in parallel with multiple attention heads, if there was only one attention mechanism, such rich information would not be available. We compute the output vector as follows:

$$Attention(K, q) = \sum_{i}^{N} m_i k_i \tag{16}$$

$$m_i = \frac{\exp\left(s\left(k_i, q\right)\right)}{\sum_{j=1}^{N} \exp\left(s\left(k_j, q\right)\right)} \tag{17}$$

$$s\left(k_i, q\right) = W_f \tanh\left(\left[k_i; q\right]\right) + b_f \tag{18}$$

s is the alignment function for learning semantic relevance.

$$u = MHA\left(h_i^L, \left[o^{t-1} v_a\right]\right) = \left(u^1 \oplus u^2 \oplus \ldots \oplus u^{\text{head}}\right) \cdot W_h \tag{19}$$

$$u^h = Attention^h\left(h_i^L, \left[o^{t-1}; v_a\right]\right) \tag{20}$$

where o^{t-1} is the contextual representation of the attention output of the upper layer. u^h is the output of the h-th attention function and head is the number of parallel attention functions.

3.7.2 Self-attention

The self-attention mechanism makes it possible to learn the correlation between the current word and the words in the previous part of the sentence and to further explore the word dependencies between sentences.

$$e = Attention(u, u) \tag{21}$$

3.7.3 Full connected layer

We use a fully connected layer to update the context representation to strengthen the context representation for a given aspect, as follows:

Table 2 The detailed Statistics of experimental datasets

dataset	Positive		Neutral		Negative	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	169	870	128
Twitter	1561	173	3127	346	1560	173

$$o^a = sigmoid\left(W_o o^{a-1} + W_e e\right) \tag{22}$$

where W^o and W^e are learnable weight matrices. $a \in [1, A]$ is the location of the present layer , a is the number of attention layers and *sigmoid* is nonlinear activation function.

3.8 Output layer

We use the *softmax* function to get the probability distribution p of aspect word sentiment polarity.

$$p = softmax\left(W_p o^a + b_p\right) \tag{23}$$

3.9 Model training

The model introduces cross-entropy and L_2 regularization as loss functions, as follows:

$$LOSS = \sum_{j=1}^{|P|} \widehat{p}_j \log p_j + \lambda \|\theta\|^2 \tag{24}$$

where P is the classification category set and $|P|$ is the number of classification categories. λ is the parameter of regularization and θ denotes all trainable parameters.

4 Experiments

4.1 Datasets

The experiments in this paper were conducted on the Sem-Eval 2014 Task4 dataset [2] and the ACL2014 Twitter dataset [30] collected by Dong et al. The Sem-Eval 2014 dataset includes user comments from two domains, Laptop and Restaurant, and the third dataset is a data sample whose sentiment polarity contains positive, neutral, and negative, with the conflicting data samples removed. The number of training and testing samples for each sentiment polarity on different datasets is shown in Table 2.

Table 3 Overall performance of different models

Models	Laptop		Restaurant		Twitter	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
SVM-feature	70.49	–	80.16	–	63.40	63.30
TD-LSTM	68.13	–	75.63	–	70.80	69.00
ATAE-LSTM	68.70	–	77.20	–	–	–
MemNet	70.64	65.17	79.61	69.64	74.48	69.90
IAN	72.05	67.38	79.26	70.09	72.50	70.81
AEN	73.51	69.04	80.89	72.14	72.83	69.81
PBAN	74.12	–	81.16	–	–	–
ASGCN	75.55	71.05	80.77	72.02	72.15	70.40
MHAGCN	75.85	71.38	81.43	73.15	73.03	70.96
MHAGCN(BERT)	**79.06**	**75.70**	**82.57**	**75.83**	**74.53**	**73.75**

4.2 Experimental settings

In the experiments, we have continuously adjusted the experimental data of the model to obtain the optimal hyper-parameters. For the GloVe embedding, the dimension of the word vector is 300; for the BERT embedding, the dimension of the word vector and the dimension of the hidden state is 768. We use the Adam optimizer in our model, and the learning rate is set to 2×10^{-5}. To prevent the effect of over-fitting, the dropout rate is set to 0.1, batch size is 64 and the L_2 regularization is 1×10^{-5}. Through continuous optimization of the experimental parameters, we found that the best experimental results were obtained when the number of GCN layers was 2 and the number of hierarchical multi-head attention layers was 3. We implement our proposed model using Pytorch. We adopt two evaluation metrics to assess the model performance: Accuracy and Macro-F1.

4.3 Baseline models

To further show the performance of the model, we compared the proposed model with several baseline models and some state-of-the-art models are shown in Table 3. The two best experimental results from the three datasets are shown in bold.

SVM-feature [17] is a traditional support vector machine-based model with extensive feature engineering by using n-gram features, analytical features and dictionary features for aspect-based sentiment classification.

TD-LSTM [22] models the context in front of and behind the aspect words, using the context in both directions as feature representation. It uses two LSTMs, and then, the hidden state vectors of the last time step of the two LSTMs are stitched together and fed into softmax for classification. Thus, the result of sentiment classification is obtained.

ATAE-LSTM [31] proposes the model of attention-based LSTM with aspect embedding and enhances the model by learning the hidden relations between the context and aspect to acquire sentiment classification results.

MemNet [32] uses multiple attention layers on word embedding, using context as external memory, and calculates the attention expressions of each layer as input for the next layer to recompute.

IAN [23] models aspect and context separately and uses the attention mechanism to link the two. The attention mechanism added when modeling aspect uses context as the query vector, and when modeling context uses aspect as the query vector, so that the interaction between the two is achieved.

AEN [33] designs attention encoding networks to interact with aspect words and contexts and adds label smoothing canonical terms to the loss function. In addition to utilizing the Glove embedding, the model uses a pre-trained BERT model.

PBAN [34] adds location information to word embedding and then processes aspect embedding and context embedding through BiGRU to obtain hidden states, respectively. Through the bidirectional attention mechanism, the correlation between aspects and sentences is analyzed.

ASGCN [15] is a model based on GCN for aspect-specific sentiment classification, starting with a bidirectional long short-term memory network layer to capture contextual information about word order and adding a multi-layer graph convolutional structure after the LSTM output.

4.4 Experimental results

This section presents the experimental results of the proposed model approach and other state-of-the-art methods on three datasets and their analysis. The accuracy of our model on the Laptop, Restaurant and Twitter datasets is 79.06, 82.57 and 74.53%, respectively, and the Macro-F1 values were 75.70, 75.83 and 73.75% on the Laptop, Restaurant and Twitter datasets, respectively. The experimental results are shown in Table 3, and the best results are in bold. We can see from the table that the performance of our model is consistently higher than the performance of other comparative models. Because of the simple structure of LSTM, the classification accuracy is the lowest among all methods. It cannot distinguish between aspects and other words used in the context and even ignore the aspect information. Therefore, it does not use target information. ATAE-LSTM has a higher performance than TD-LSTM. TD-LSTM uses the location of the target to divide the context as left context and right context; and use standard LSTM to process the target. In this approach, the goals are more centralized. ATAE-LSTM integrates the attention mechanism with LSTM to get more important context information for disparate aspects, which attended great experimental results. MemNet has a strong capability in aspect sequence modeling, but context and sequence information is lacking. IAN models aspect words and contexts, and context and aspect interactions are accomplished using two attention mechanisms, and the model is able to focus on those words that have a significant impact on determining affective polarity outcomes. Compared

Table 4 Overall ablation results

Ablation	Laptop		Restaurant		Twitter	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
w/o GCN	73.8	69.52	79.16	72.32	72.58	70.32
w/o Conv	74.25	70.76	81.35	72.97	72.69	69.96
w/o MHA	74.32	69.39	70.65	71.83	72.11	69.89
MHAGCN	75.85	71.38	81.43	73.15	73.03	70.96

with the approach proposed in this paper, MemNet and IAN are still not effective enough, probably because their aspect and context interactions are coarse-grained, which may lead to the loss of interaction information.

Our model has a significant improvement over the AEN model, which accomplishes the interaction between context and aspect words and extracts semantic features through a multi-headed attention mechanism. However, the experimental performance of our approach is better because the attention mechanism in not able to obtain the distant dependency information. PBAN uses location information to calculate the relative distance between each context word and the relevant context, and combines location information with a bidirectional attention mechanism, and its performance is better than IAN, which shows that introducing location information can also improve model performance. ASGCN uses GCNs to extract syntactic relations from the output of Bi-LSTM and employs an attention mechanism to exploit the syntactic relations in the input sentences to enrich the aspect-level contextual representation.

Experimental comparative analysis with these baseline models shows that our model performance has some improvement effect on all three datasets. Our approach incorporates syntactic dependency information and focuses on the interaction between aspect words and contexts through a hierarchical multi-headed attention mechanism.

4.5 Ablation analysis

As shown in Table 4, the performance of the MHAGCN model is better than the experimental results of these several ablation models, which represents that these components are essential in our proposed model.

We removed the GCN mechanism based on the dependency tree, and we called this ablation model "w/o GCN". The experimental data can be seen to drop significantly on the datasets Restaurant and Laptop, with insignificant changes on the Twitter dataset. This is because the data on the Twitter dataset are biased toward colloquialism, with less pronounced syntactic information and less sensitive to emotional dependency relationships.

We removed the convolution layer, namely "w/o Conv" and the result is completely lower than the MHAGCN model, but the change is not significant. This

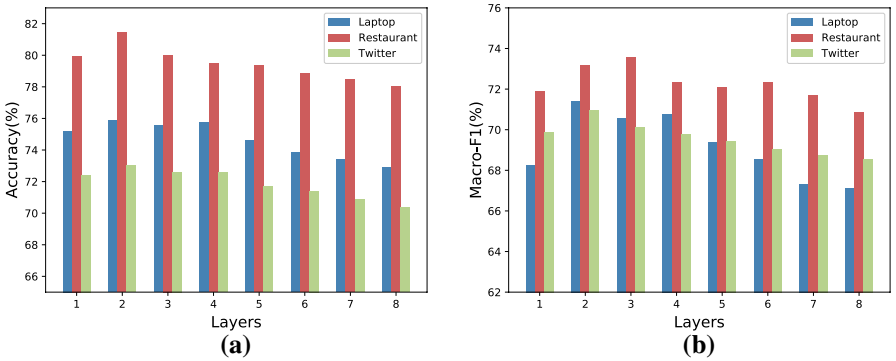

Fig. 4 Effect of GCN layers in Accuracy and Macro-F1

experiment shows that this ablation model has a relatively small effect on the overall experimental results, but it is an indispensable part.

We remove the hierarchical multi-headed attention mechanism in MHAGCN and replace it with the attention layer in MemNet, called "w/o MHA". The experimental results are significantly lower than our model, and our model can effectively prevent the loss of aspect information.

4.6 Effect of GCN layers

In this section, we increase the number of GCN calculation layers from 1 to 8 with the other parameters unchanged to explore the influence of calculating the GCN layers on the sentiment classification ability of the model. We recorded Accuracy and Macro-F1 on two datasets, as shown in Fig. 4.

As the number of layers increases, the performance first increases and then decreases. It is obvious from the figure that the original performance is low, but gradually improves with increasing depth. When the second layer is reached, our model achieves the best performance. The decrease in performance may be due to the increasing number of layers, which makes it difficult to train the model and overfitting occurs.

4.7 Effect of attention layers

In that section, we increase the number of attention layers from 1 to 8 to evaluate the effect of the number of attention layers on model performance in a hierarchical multi-headed attention layer. We are able to conclude from Fig. 5 that the best performance is achieved when the attention layer reaches the third layer. When the number of attention layers is 1, the performance is lower than that of the model with 2 attention layers on the Restaurant and Laptop datasets, but better on the Twitter dataset. So when the attention layer is 1, there is a deficiency in learning a more

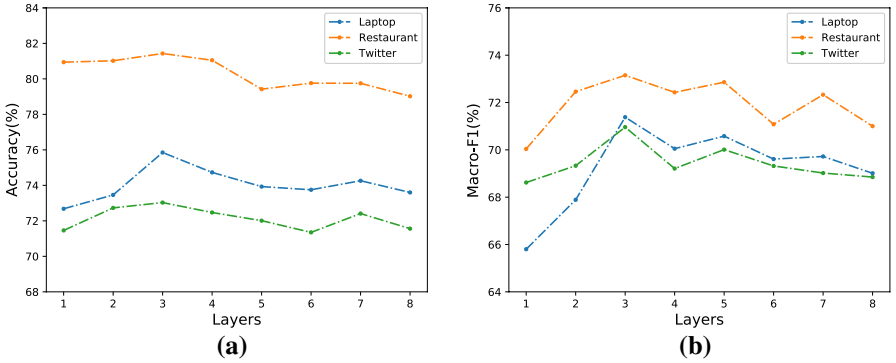

Fig. 5 Effect of attention layers in Accuracy and Macro-F1

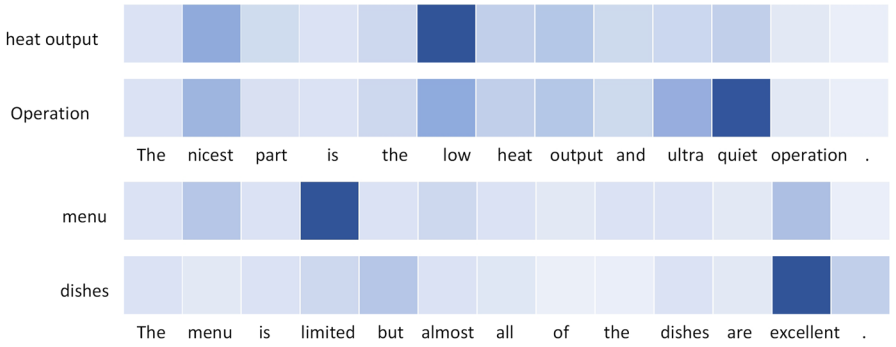

Fig. 6 Two cases with visualized attention weights assigned by MHAGCN

comprehensive contextual representation, and the Restaurant and Laptop datasets are more sensitive to changes in the number of attention layers. As the number of layers increases, it increases the complexity of the model, making the performance decrease.

4.8 Case study

In order to understand our model in better way and visualize which words are determining the sentiment polarity of a given aspect word in a sentence, we give several cases for visual analysis to calculate the attention weights of the words as shown in the Fig. 6. Darker colors represent higher scores.

The first example is "The nicest part is the low heat output and ultra quiet operation. For the aspect "heat output", the model MHAGCN gives more attention to "low"than"output"; for the aspect "operation", the context word "quiet"is given the most important attention. The second example is "The menu is limited, but almost all of the dishes are excellent. The sentence contains two aspects: "menu" and "dishes", and the affective polarity of the aspect "menu" is negative,

while the affective polarity of the aspect “dishes” is positive. In the figure, we can see that the model gives the highest attention to the contextual word "limited" for the aspect “menu” and to the contextual word “dishes”. The most important attention is given to the context word “excellent”. It can be seen that when the text contains multiple aspect words, the model MNAGCN can correctly identify the opinion words related to them and give the corresponding attention weights, and can accurately identify the different sentiment words of aspect words with different sentiment polarity in the sentence.

5 Conclusion

Aspect-level sentiment classification is a relatively popular direction in the field of natural language processing. In this paper, we propose an ALSC neural network approach based on graph convolutional networks and a hierarchical multi-head attention mechanism. Specifically, we first use the multi-head self-attention mechanism and convolutional layer to obtain the context hidden state, secondly employ the dependency tree-based graph convolutional network to capture the syntactic dependency information, and finally use the hierarchical multi-head attention mechanism to establish the relationship between aspect words and context to realize the interaction between them. Our proposed method can effectively combine syntactic information and semantic relations to better predict the sentiment polarity of aspect words. We obtained excellent results from extensive experiments on three datasets, which implies that it is effective and feasible to improve the performance of sentiment prediction by using sentence structure information and semantic information.

In future work, we will consider further improvements to the model MHAGCN. Since short textual comments usually omit a large amount of background commonsense knowledge, it is difficult to infer the true sentiment polarity only from the text itself, so we introduce commonsense knowledge and dependency types into the model to improve the performance of the model.

References

1. Kang H, Yoo SJ, Han D (2012) Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. Expert Syst. Appl. 39(5):6000–6010. https://doi.org/10.1016/j.eswa.2011.11.107
2. Pontiki M, Galanis D, Pavlopoulos J, Papageorgiou H, Androutsopoulos I, Manandhar S(2014) SemEval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 27– 35. Association for Computational Linguistics, Dublin, Ireland
3. Schouten K, Frasincar F (2016) Survey on aspect-level sentiment analysis. IEEE Trans. Knowl. Data Eng. 28(3):813–830. https://doi.org/10.1109/TKDE.2015.2485209
4. Tay Y, Tuan L.A, Hui S.C(2018) Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,(AAAI-18), pp. 5956– 5963

5. Tang D, Qin B, Liu T(2015) Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422– 1432. Association for Computational Linguistics, Lisbon, Portugal
6. Zainuddin N, Selamat A, Ibrahim R (2018) Hybrid sentiment classification on twitter aspect-based sentiment analysis. Appl. Intell. 48(5):1218–1232. https://doi.org/10.1007/s10489-017-1098-6
7. Marcheggiani D, Täckström O, Esuli A, Sebastiani F (2014) Hierarchical multi-label conditional random fields for aspect-oriented opinion mining. In: de Rijke M, Kenter T, de Vries AP, Zhai C, de Jong F, Radinsky K, Hofmann K (eds) Advances in Information Retrieval. Springer, Cham, pp 273–285
8. Mikolov T, Zweig G(2012) Context dependent recurrent neural network language model. In: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 234– 239 https://doi.org/10.1109/SLT.2012.6424228
9. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput. 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
10. Gers FA, Schmidhuber J, Cummins FA (2000) Learning to forget: Continual prediction with LSTM. Neural Comput. 12(10):2451–2471. https://doi.org/10.1162/089976600300015015
11. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y(2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724– 1734. Association for Computational Linguistics, Doha, Qatar
12. Kim Y(2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar
13. De Luca P, Galletti A, Giunta G, Marcellino L (2020) Accelerated gaussian convolution in a data assimilation scenario. In: Krzhizhanovskaya VV, Závodszky G, Lees MH, Dongarra JJ, Sloot PMA, Brissos S, Teixeira J (eds) Computational Science - ICCS 2020. Springer, Cham, pp 199–211
14. Bo D, Wang X, Shi C, Shen H(2021) Beyond low-frequency information in graph convolutional networks. CoRR **abs/2101.00797**
15. Zhang C, Li Q, Song D(2019) Aspect-based sentiment classification with aspect-specific graph convolutional networks. 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), pp. 4568– 4578. Association for Computational Linguistics, Hong Kong, China
16. Akhtar MS, Ekbal A, Cambria E (2020) How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes]. IEEE Comput Intell Mag 15(1):64–75. https://doi.org/10.1109/MCI.2019.2954667
17. Kiritchenko S, Zhu X, Cherry C, Mohammad S (2014) NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp. 437– 442. Association for Computational Linguistics, Dublin, Ireland
18. Xue W, Li T (2018) Aspect based sentiment analysis with gated convolutional networks. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 2514– 2523. Association for Computational Linguistics, Melbourne, Australia
19. Ruder S, Ghaffari P, Breslin JG (2016) A Hierarchical Model of Reviews for Aspect-Based Sentiment Analysis. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 999– 1005. Association for Computational Linguistics, Austin, Texas
20. Zhang M, Zhang Y, Vo D (2016) Gated Neural Networks for Targeted Sentiment Analysis. In: Schuurmans D, Wellman MP (Eds.) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp 3087– 3093
21. Wang Y, Huang M, Zhu X, Zhao L (2016) Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 606– 615. Association for Computational Linguistics, Austin, Texas
22. Tang D, Qin B, Feng X, Liu T (2016) Effective LSTMs for Target-Dependent Sentiment Classification. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers, pp 3298– 3307. The COLING 2016 Organizing Committee, Osaka, Japan
23. Ma D, Li S, Zhang X, Wang H (2017) Interactive attention networks for aspect-level sentiment classification. In: Sierra C (Ed.) Proceedings of the twenty-sixth international joint conference on artificial intelligence, pp 4068– 4074

24. Fan F, Feng Y, Zhao D (2018) Multi-grained attention network for aspect-level sentiment classification. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 3433– 3442. Association for Computational Linguistics, Brussels, Belgium
25. Chen P, Sun Z, Bing L, Yang W (2017) Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 452– 461. Association for Computational Linguistics, Copenhagen, Denmark
26. Cai H, Tu Y, Zhou X, Yu J, Xia R (2020) Aspect-category based sentiment analysis with hierarchical graph convolutional network. In: Scott D, Bel N, Zong C (Eds.) Proceedings of the 28th international conference on computational linguistics, pp 833– 843
27. Zhang M, Qian T (2020) Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: Webber B, Cohn T, He Y, Liu Y (Eds.) Proceedings of the 2020 conference on empirical methods in natural language processing, pp 3540– 3549
28. Pennington J, Socher R, Manning C (2014) GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532– 1543. Association for Computational Linguistics, Doha, Qatar
29. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (Long and Short Papers), pp. 4171– 4186. Association for Computational Linguistics, Minneapolis, Minnesota
30. Dong L, Wei F, Tan C, Tang D, Zhou M, Xu K (2014) 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), pp. 49– 54. Association for Computational Linguistics, Baltimore, Maryland
31. Wang S, Mazumder S, Liu B, Zhou M, Chang Y (2018) Target-sensitive memory networks for aspect sentiment classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp. 957– 967. Association for Computational Linguistics, Melbourne, Australia
32. Tang D, Qin B, Liu T (2016) Aspect level sentiment classification with deep memory network. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 214–224. Association for Computational Linguistics, Austin, Texas
33. Song Y, Wang J, Jiang T, Liu Z, Rao Y (2019) Attentional encoder network for targeted sentiment classification. CoRR abs/1902.09314
34. Gu S, Zhang L, Hou Y, Song Y (2018) A Position-Aware Bidirectional Attention Network for Aspect-Level Sentiment Analysis. In: Proceedings of the 27th international conference on computational linguistics, pp 774– 784. association for computational linguistics, Santa Fe, New Mexico, USA