Journal of Computer and Communications, 2024, 12, 191-207
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Aspect-Level Sentiment Analysis Incorporating Semantic and Syntactic Information

Jiachen Yang[1], Yegang Li[1], Hao Zhang[1], Junpeng Hu[2], Rujiang Bai[2]

[1]School of Computer Science and Technology, Shandong University of Technology, Zibo, China
[2]Institute of Information Management, Shandong University of Technology, Zibo, China
Email: yangjc_98@163.com

How to cite this paper: Yang, J.C., Li, Y.G., Zhang, H., Hu, J.P. and Bai, R.J. (2024) Aspect-Level Sentiment Analysis Incorporating Semantic and Syntactic Information. *Journal of Computer and Communications*, **12**, 191-207.
https://doi.org/10.4236/jcc.2024.121014

Received: December 29, 2023
Accepted: January 28, 2024
Published: January 31, 2024

Abstract

Aiming at the problem that existing models in aspect-level sentiment analysis cannot fully and effectively utilize sentence semantic and syntactic structure information, this paper proposes a graph neural network-based aspect-level sentiment classification model. Self-attention, aspectual word multi-head attention and dependent syntactic relations are fused and the node representations are enhanced with graph convolutional networks to enable the model to fully learn the global semantic and syntactic structural information of sentences. Experimental results show that the model performs well on three public benchmark datasets Rest14, Lap14, and Twitter, improving the accuracy of sentiment classification.

Keywords

Aspect-Level Sentiment Analysis, Attentional Mechanisms, Dependent Syntactic Trees, Graph Convolutional Neural Networks

1. Introduction

The Internet has become inseparable from people's daily life, and people's subjective comments on popular events have become an important way to express their opinions and emotions. Sentiment Analysis (SA) is a hot research direction in natural language processing. The process of analyzing, generalizing and summarizing comments with subjective emotional tendencies is called sentiment analysis [1]. Depending on the text granularity of sentiment analysis, it can be categorized into: document-level, sentence-level, and aspect-level [2]. Among these, Aspect-based Sentiment Analysis (ABSA) aims to determine the sentiment polarity of a given aspectual term in a sentence: positive, negative, neutral.

Document-level and sentence-level sentiment analysis is only for the overall sentiment polarity of the whole sentence or document, which is not suitable for the case where the sentence contains different sentiments. As shown in **Figure 1**, the example sentence contains (in red) two "aspects": "food" and "environment". "Food" has a negative affective polarity; the aspect "environment" has a positive affective polarity. It can be seen that the goal of ABSA is to analyze attitudes or affective tendencies towards a certain entity, which is important for predicting future activities or guiding public opinion in order to make targeted decisions.

The key to the ABSA task is to obtain and establish the dependencies between aspect words and their corresponding emotion words from the context. Previous studies have mainly used the attention mechanism to model the correlation between aspect words and context, but due to the complexity of sentence structure, on the one hand, the attention mechanism is affected by the noise in the sentence, *i.e.*, words that are not related to the aspect words, and on the other hand, the attention does not blend the syntactic information between the aspect words and the emotion words well.

Given that the syntactic structural information between aspect words and their corresponding sentiment words can provide help for sentiment polarity prediction, the method of constructing Graph Neural Network (GNN) based on dependency trees has become an important direction of ABSA research in recent years [3]. This type of research parses sentences into dependency syntax trees that express whether there are dependencies between different words and can be directly used in Graph Convolution Network (GCN). This type of model utilizes the syntactic structure information of the sentence, but if there is no direct syntactic relationship between the aspect words and sentiment words in the sentence, the performance of the model will be affected. For example, in **Figure 1**, there is a dependency between "food" and the emotion word "good", and there is also a dependency between "environment" and "good", if we only consider the dependency relationship, the model will get wrong judgment, and for the relationship between the aspect word "food" and the word "not", we need to obtain the relationship through two layers of GCN. The relationship between the aspect word "food" and the word "not" needs to be obtained through two layers of GCN, which will also make the aspect word "environment" obtain the dependency relationship with the noise word "not". Therefore, it is not enough to utilize only the structural information of the dependency syntactic tree, but the semantic information of the aspect word and the context should be fully considered in the

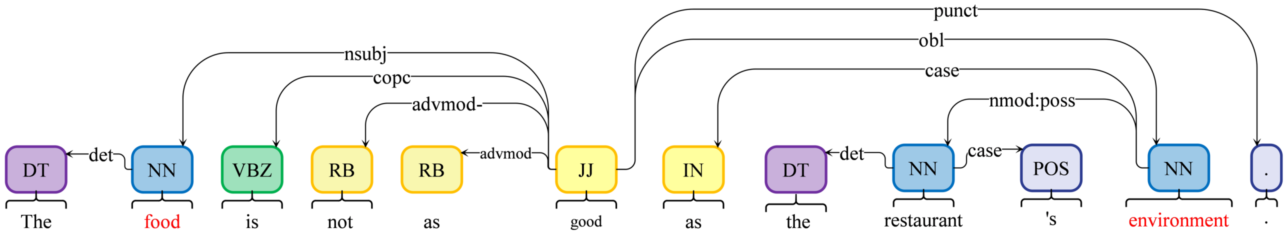

Figure 1. An example with a dependency tree.

sentiment analysis.

In order to fully utilize the sentence structure information, Zhang *et al.* [4] first proposed to construct a model of GCN using dependent syntactic trees, obtaining syntactic structure through dependent syntactic relations, and mining syntactic information by using GCN. Sun *et al.* [5] utilized BiLSTM to provide the original node feature vectors, and augmented the feature vectors learned by BiLSTM by using GCN, and used syntactic information as the GCN edge information for modeling. Zhang *et al.* [6] argued that previous models did not consider different types of syntactic dependencies and proposed a new architecture that uses a global lexical graph to encode corpus-level word co-occurrence information, builds concept hierarchies on both syntactic and lexical graphs for distinguishing different dependencies, and finally makes full use of both graphs by using a two-layer interaction graph convolutional network.

In order to enhance the semantic or structural information, Xu *et al.* [7] proposed multi-head self-attention and multi-head interactive attention, which capture the contextual information through the attention coding layer, integrate the syntactic information through the attention-enhanced graph convolution, and finally integrate the information using the multi-head interactive attention to obtain the final feature representations. Wang *et al.* [8] put the word positional distance feature into graph convolutional network, and at the same time, utilize the textual word syntactic distance features to weight the adjacency matrix of the graph convolutional network, and designed semantic interaction and syntactic interaction to deal with the semantic and syntactic information between words respectively. Cui *et al.* [9] constructed a syntactic graph attention network based on the dependency syntactic tree, with the distinction of the importance of the real line dependency relationship to effectively establish the relationship between the target word and the sentiment word, and constructed a global graph attention network to mine the target and the sentiment words' missing relationship to further improve the model performance. Zhu *et al.* [10] constructed text sequence graph and augmented dependency graph to utilize the rich structural information, meanwhile, proposed two kinds of attention networks to learn the sentence representation from different perspectives, and merged the above two kinds of information to get the final representation of the sentence.

However, the above methods usually only enhance the semantic representation of sentences or allow the model to learn richer information about dependencies, but neglect the effective fusion of syntactic structural information and semantic information, and the model is unable to take into account both syntactic structural information and semantic information. In this paper, two attention mechanisms, multi-head attention and multi-head self-attention, are used to learn different semantic information of a sentence, while the dependent syntactic tree provides syntactic structure information for the model, which effectively fuses the two different kinds of information and improves the performance of the model. A graph convolutional network model (GCN combines semantic and syntactic structure information (CSSGCN)) is proposed, and the CSSGCN model utilizes two different attention mechanisms to capture different semantic information respectively, and the aspect word attention mechanism captures the aspect word with its context and the self-attention mechanism captures global semantic information. Unlike the models proposed in previous studies, this model enhances the traditional GCN model by representing the semantic information captured by the two attention mechanisms as an attention score matrix, which is fused with a neighbor matrix with sentence dependencies and structural information. Finally, aspect-specific features for aspect word sentiment classification are obtained using multilayer graph convolution operations.

2. Related Technologies

2.1. Dependent Syntactic Tree

Dependency syntax was first proposed by the French linguist L. Tesniere, which analyzes a sentence as a dependency syntactic tree, describing the dependencies between individual words. These dependencies express the semantic dependencies between the components of the sentence. The dependencies between all words form a syntactic tree, and the root node of the tree is the core predicate of the sentence, which is used to express the core content of the whole sentence. By depending on the dependencies in the syntactic tree, two words with a specific syntactic relationship can be obtained. Two lexemes with a dependency relation are not necessarily adjacent to each other, and there are often other lexemes between the two lexemes.

Dependency relations are represented by a core word (head) and a dependent word, each core word corresponding to the center of its constituent (e.g., a noun to a noun phrase, a verb to a verb phrase). The most commonly used relations fall into two main categories: subordinate clause relations and modifier relations.

Dependency structures are mostly interconnected, have a specified root node, and are either acyclic or planar, so a dependency tree is a directed graph with a single specified root node that has no incoming arcs, every node except the root node has one and only one incoming arc, and every node from the root node to every node has one and only one a path from the root node to each node. Since dependent syntax results in a directed graph, where words are nodes and directed edges between words represent the syntactic relationships between them, existing research tends to utilize methods based on graph structures, such as graph convolutional neural networks.

2.2. Graph Convolutional Neural Network

Deep learning has achieved good results in many fields due to its powerful feature extraction and fitting capabilities, replacing traditional machine learning and manual feature extraction methods. However, traditional deep learning methods are usually more suitable for data represented in Euclidean space, and cannot solve the problem of non-Euclidean space well. Data in non-Euclidean space is usually represented by graphs, which represent the relationships between objects.

The graph structure is relatively complex and generally not neat; a network contains different numbers of nodes, and different nodes also contain different neighbors. This makes traditional neural network operations (e.g., convolutional operations, etc.) not work well on graph structures. Another point is that traditional machine learning methods usually assume that the samples are independent of each other, but on a graph structure, the samples are usually connected to each other. A graph structure is usually denoted by $G=(V, E)$, where V denotes the set of nodes and E denotes the set of edges. In order to apply deep learning methods to graph structures, Graph Neural Network (GNN) has been proposed. Some important neural network operations are also redefined on graph structures, such as the traditional convolution operation, where each pixel can be viewed as a point, and then weighted and summed with the surrounding points. For graph structure, a similar approach can be taken for convolution by weighting and summing the neighbors of a node on the graph as shown in **Figure 2**.

Graph Convolution Neural Networks (GCNNs) methods are categorized into two groups, spectral domain based methods and null domain based methods. Spectral domain based methods define graph convolution by introducing filters from the perspective of graph signal processing, where the graph convolution operation is interpreted as the removal of noise from the graph signal. Null domain based methods represent graph convolution as aggregating feature information from neighbors. The graph convolution neural network has four features:

1) GCN is a natural generalization of convolutional neural networks to graphs.

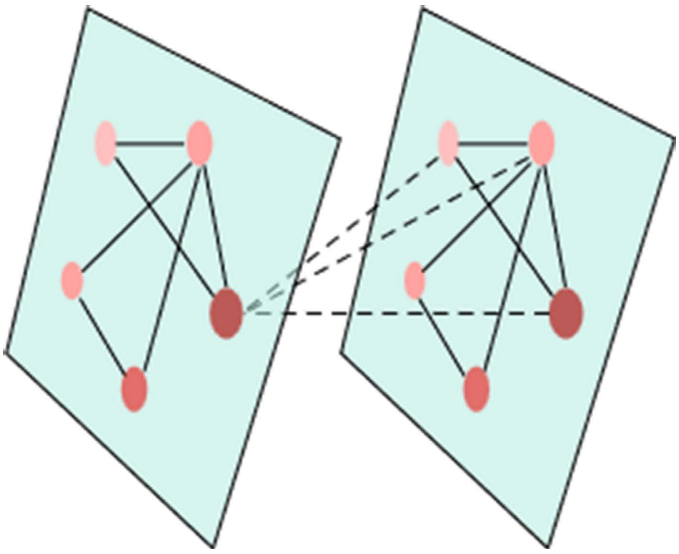

Figure 2. Nodes of the graph are updated.

2) GCN is capable of end-to-end learning of node feature information and structural information at the same time, which is currently the best choice for the task of learning graph data.

3) GCN Extremely broad applicability, applicable to nodes and graphs of any topology.

4) GCN works far better than other methods on public datasets for tasks such as node classification and edge prediction.

3. The CSSGCN Model

The CSSGCN model utilizes the aspect word attention mechanism to capture the local semantic information of the aspect word with its context; and the self-attention mechanism to capture the global semantic information. The semantic information captured by the two attention mechanisms is represented as an attention score matrix and fused with a neighbor matrix with sentence dependencies and structural information to enhance the traditional GCN model [11]. Finally, aspect-specific features for aspect word sentiment classification are obtained using multilayer graph convolution operations. The overall architecture is shown in **Figure 3**, which shows that the model is mainly composed of five parts: input and word vector embedding layer, Bi-LSTM layer, attention layer, dependency syntax layer, and graph convolution network layer.

3.1. Input and Word Vector Embedding Layer

For the preprocessed text, the model takes as input the sentence aspect pair (s,a), where s is a sentence of length n, $s=\{s_1,s_2,\cdots,s_n\}$, a is an aspect of length m, $a=\{a_1,a_2,\cdots,a_m\}$, and is also a subsequence of the sentence s, The pre-trained GloVe word vectors [12] are used to map each word in the input utterance to a low-dimensional real-valued vector to obtain the embedding matrix $E\in\mathbb{R}^{|V|\times d_{\text{emb}}}$, $|V|$ is the size of the vocabulary, and d_{emb} denotes the dimension of each word vector. Thus sentence s has the corresponding word embedding $x=\{x_1,x_2,\cdots,x_n\}$.

3.2. Bi-LSTM Layer

The sentiment expressed by words is affected by contextual information, as shown in **Figure 1**, "The food is not as good as the restaurant's environment", where "not" is the negation of "good". "LSTM can learn the information from the front to the back of the sentence during the training process, but it cannot encode the information from the back to the front, so Bi-LSTM is used to learn the hidden information in the context, which is a combination of forward and backward LSTM combined. The forward LSTM for sentence s is denoted as $\overrightarrow{\mathbf{H}^{\text{F}}}=\left\{\overrightarrow{\mathbf{h}_1},\overrightarrow{\mathbf{h}_2},\cdots,\overrightarrow{\mathbf{h}_n}\right\}$, and the backward LSTM for sentence s is denoted as $\overleftarrow{\mathbf{H}^{\text{B}}}=\left\{\overleftarrow{\mathbf{h}_1},\overleftarrow{\mathbf{h}_2},\cdots,\overleftarrow{\mathbf{h}_n}\right\}$, The corresponding vectors of $\overrightarrow{\mathbf{H}^{\text{F}}}$ $\overleftarrow{\mathbf{H}^{\text{B}}}$ are spliced to the final hidden state vector $H=\{h_1,h_2,\cdots,h_n\}$ generated by Bi-LSTM, where

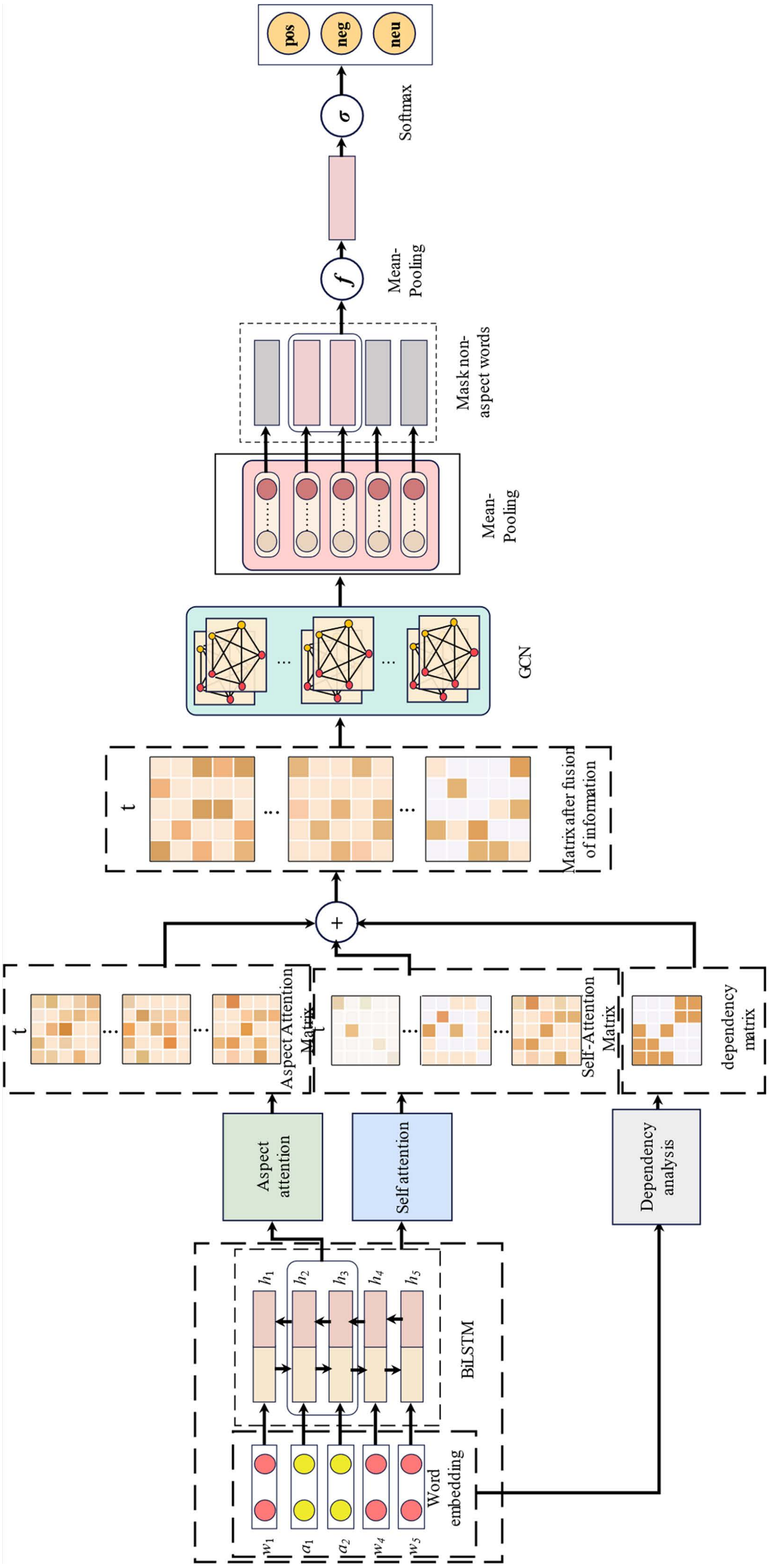

Figure 3. The overall architecture of CSSGCN.

$h_i \in \mathbb{R}^{2d}$, H contains a subsequence of the corresponding aspect word, and H is used as the initial input to the model.

3.3. The Attentional Layer

The attention mechanism captures the interaction between aspectual words and context [13], as shown in **Figure 2**, both aspectual and self-attention matrices are composed of t attention adjacency matrices, representing both are generated by the t-head attention mechanism, t is a hyperparameter, and the fusion of the two attention matrices leads to a better semantic characterization of the model.

3.3.1. Multi-Attention Mechanisms for Aspect Words

Aspect-level sentiment analysis aims to determine the sentiment polarity of a particular aspect in a contextual sentence, so it needs to model specific inter-semantic associations based on different aspect words. In this paper, we use the aspect word multi-head attention mechanism which is designed to allow the model to learn the contextual semantic information associated with the aspect words. So the vectors of hidden states of aspect words are used as query in the attention computation, and the vectors of hidden states generated by the coding layer are used as key, and both of them do the attention computation to make the model get the attention matrix of aspect words.

$$A_{\text{aspect}}^{i} = \tanh\left(H_a W^a \times \left(K W^k\right)^{\mathrm{T}} + b\right) \tag{1}$$

In Equation (1), K is equal to the hidden state vectors H, $W^a \in \mathbb{R}^{d \times d}$ and $W^k \in \mathbb{R}^{d \times d}$ generated by the coding layer, and $H_a \in \mathbb{R}^{n \times d}$ is obtained as a representation of the aspect word by average pooling of h_a and then copying it n times. In this paper, the aspectual attention mechanism of t-heads is used to obtain the attention scores of the sentences, and A_{aspect}^{i} denotes the attention matrix obtained from the i-head.

3.3.2. Self-Attention Mechanism

Similar to the multi-head aspect attention mechanism, in this paper A_{self} , it is also possible to construct a self-attention mechanism with t heads, which captures the semantic information between any two words in a single sentence, and the hidden state vectors outputted by the coding layer provide the *query* and *key* for the attention computation:

$$A_{\text{self}}^{i} = \frac{Q W^Q \times \left(K W^K\right)^{\mathrm{T}}}{\sqrt{d_k}} \tag{2}$$

Q and K in Equation (2) are derived from the hidden vector H produced by the coding layer, $W^Q \in \mathbb{R}^{d \times d}$ and $W^K \in \mathbb{R}^{d \times d}$ are the weights that can be learned, and d_k is the dimensionality of the vector obtained after the KW^K calculation.

3.4. Dependent Syntax Layer

A dependency syntax tree can be interpreted as a graph G with n nodes, where nodes denote words in a sentence and edges denote syntactic dependency paths between words in the graph. The dependency tree G of any sentence can be represented as an $n \times n$ adjacency matrix *D*. If node i is connected to node j through a single dependency path in *G*, then $D_{ij} = 1$, otherwise $D_{ij} = 0$, as shown in Equation (3):

$$D_{ij} = \begin{cases} 1, & \text{Nodes } i,\ j \text{ have dependencies} \\ 0, & \text{Nodes } i,\ j \text{ do not have dependencies} \end{cases} \tag{3}$$

In the previous section, the *t*-head attention mechanism can obtain *t* attention matrices, so we copy the matrix D *t* times so that the number of dependent syntactic adjacency matrices is the same as the number of attention matrices as shown in Equation (4):

$$D = \left\{ D^1, \cdots, D^t \right\} \tag{4}$$

We fused the multi-head attention score matrix focusing on aspects, the multi-head self-attention score focusing on global semantic information, and the adjacency matrix with dependent syntactic structural information to obtain a matrix A^i rich in semantic and syntactic structural information, as shown in Equation (5) as follows.

$$A^i = \text{softmax}\left(A^i_{\text{aspect}} + A^i_{\text{self}} + D^i \right) \tag{5}$$

Matrix $A^i \in \mathbb{R}^{n \times n}$ based on fusing semantic information syntactic structure information.

3.5. Graph Convolutional Network Layer

The dependent syntax layer produces t different matrices $A \in \mathbb{R}^{t \times n \times n}$ rich in semantic and syntactic information, so t graph convolution operations are required in each layer of the graph convolution network. If h^{l-1} is taken as the input state, and h^l is denoted as the layer *l* output state, then h^0 is denoted as the output after the sentence encoding layer (Bi-LSTM). Each node in the *l*th layer GCN is updated according to the hidden representation of its neighborhood:

$$h_i^l = \sigma\left(\sum_{j=1}^{n} A_{ij} W^l h_j^{l-1} + b^l \right) \tag{6}$$

In Equation (6) h_i^l denotes the hidden state representation of node *i* in the *l*th layer GCN, b^l denotes the bias of the *l*th layer GCN, W^l denotes the linear change weight, which is a learnable parameter matrix, and σ denotes the nonlinear activation function, and A_{ij}, which is enriched with semantic-syntactic information, will be used as the adjacency matrix of the GCN. The final output of the *l*th layer GCN is denoted as hl, as shown in Equation (7):

$$H^l = \left\{ h_1^l, h_2^l, \cdots, h_n^l \right\} \tag{7}$$

After aggregating the node information at each layer of GCN, the final feature representation of the word is obtained. The non-aspect word representations in the feature representations output by the GCN are masked to obtain the aspect word feature representations, and the aspect word feature representations h_a^l retained most of the aspect information through average pooling.

$$h_a^l = f\left(h_{a_1}^l, h_{a_2}^l, \cdots, h_{a_m}^l\right) \tag{8}$$

The $f(\cdot)$ in Equation (8) represents the average pooling function used to augment the aspect representation through the GCN layer. The representation h_a^l of aspect words is fed into the linear layer after passing through the Softmax function in order to obtain the sentiment classification probability.

$$c(a) = \text{softmax}\left(W_c h_a^l + b_c\right) \tag{9}$$

In Equation (9) W_c and b_c are learnable parameters.

3.6. Train

The cross-entropy loss function is used as the loss function in this paper:

$$L(\theta) = -\sum_{(s,a)\in\mathcal{D}} \sum_{c\in\mathcal{C}} \log c(a) \tag{10}$$

The cross entropy as loss function benefit is that using the sigmoid function in gradient descent avoids the problem of reduced learning rate of the mean square error loss function.

In Equation (10), $\mathcal{D}$ contains all pairs of aspects of a sentence, *a* denotes the aspect that occurs in the sentence. θ denotes all trainable parameters, and $\mathcal{C}$ is the set of sentiment polarities.

4. Experimentation and Analysis

4.1. Datasets

In this paper, experiments were conducted on three publicly available datasets based on fine-grained sentiment analysis including: the SemEval2014Task 4 restaurant review dataset [14] Rest14 and the laptop review dataset Lap14, and the Twitter tweet review dataset [15]. Each dataset is composed with real reviews, and each review aspect word has agyhg5 sentiment polarity corresponding to it, and the sentiment polarity includes: positive, neutral and negative. In this paper, Sun *et al.* preprocessing method is adopted to process the datasets, and the statistical results of each dataset after processing are shown in **Table 1**.

4.2. Comparison Experiment

In order to evaluate the performance of the CSSGCN model, this paper compares the model baseline model with some models with advanced performance.

ATAE-LSTM [16]: this model uses information about the context and aspectual words in a sentence for splicing and modeling through an attention mechanism.

ASGCN: This model proposes to model GCN based on dependency trees to learn the syntactic information of sentences and word dependencies.

Table 1. Statistics for the three experimental datasets.

Datasets	Clusters	Aggressive	Neutral	Negativity
Rest14	Train	2164	637	807
	Test	727	196	196
Laptop	Train	976	455	851
	Test	337	167	128
Twitter	Train	1507	3016	1528
	Test	172	336	169

CDT: This model learns sentence feature representations via BiLSTM, which learns sentence-wise feature representations via dependency tree-based GCN convolution.

BiGCN: This model constructs a grammar graph based on dependency trees constructs a vocabulary graph based on word co-occurrence relations and designs an interactive graph convolutional network to learn node features.

AEGCN: This model represents the text in a two-channel form utilizing a multi-head attention mechanism as well as an improved GCN based on dependency trees, and enhances the representation using interactive attention.

MIGCN: This model employs a Multi-Interaction Graph Convolutional Network for the fusion of semantic and syntactic features, while utilizing semantic information to complement syntactic structure and address the noise from dependent parsing.

DGAT: The model uses a syntactic graph attention network to realize the differentiation of the importance of dependent syntactic relations, to more effectively establish the relationship between aspect words and emotion words, and thus to obtain a more accurate representation of emotion features.

SEDC-GCN [13]: this model proposes two kinds of specific attention, one semantic-specific and the other aspect-specific structural, to learn sentence representations from two different perspectives.

In Table 2, from the experimental results, it can be seen that the CSSGCN model is more effective than the model modeled by just using LSTM alone or the attention mechanism (ATAE-LSTM), indicating that using only the attention mechanism ignores the syntactic structural information, and the model is less capable of recognizing for the case where the context and aspectual words are far away from each other. Some models using GCN and GAT (CDT, TD-GAT, BiGCN, etc.) use graph convolution or graph attention networks that can capture dependencies on long-distance words in the context, but lack information between context and aspectual words. For enhancing semantic information or dependencies graph convolution models (AEGCN, MIGCN, DGAT, SEDC-GCN) improve the accuracy of the model but do not interactively fuse syntactic structural information with semantic information the model effect is limited.

CSSGCN's model accuracy and F1 values in the three publicly available datasets are better than or close to those of the other models. CSSGCN makes full use of semantic and syntactic structural information, which allows the model to learn information that is useful for the final sentiment categorization, and thus improves the model accuracy.

4.3. Ablation Experiment

In order to further verify the effectiveness of each module on the model, the results of the ablation experiments on the model in this paper are shown in Table 3:

1) w/o-A_s: removal of self-attention mechanisms.

2) w/o-A_a: Removing aspectual attention mechanisms.

3) w/o-D_t: removal of syntactic structure information.

Firstly, the CSSGCN model is considered as the baseline model, and it is shown in Table 3 that removing the self-attention mechanism reduces the performance of the model, which verifies the importance of the global information of the sentence for aspect-level sentiment analysis. If the attention mechanism for aspectual words is removed, it shows that the model lacks the ability to capture aspectual semantics, resulting in 0.93%, 1.73%, and 1.63% lower accuracy for Res14, Laptop, and Twitter, respectively. This indicates that capturing the

Table 2. Experimental results.

Models	Res14		Laptop		Twitter	
	Acc	F1	Acc	F1	Acc	F1
ATAE-LSTM	77.20	-	68.70	-	-	-
ASGCN	80.77	72.02	75.55	71.05	72.15	70.40
CDT	82.30	74.02	77.19	72.99	74.66	73.66
BiGCN	81.97	73.48	74.59	71.84	74.16	73.35
AEGCN	81.43	73.66	75.91	71.88	73.86	71.59
MIGCN	82.32	74.31	76.59	72.44	73.31	72.12
DGAT	82.68	75.53	76.49	72.75	73.99	72.53
SEDC-GCN	83.30	77.51	77.74	74.68	74.42	73.37
CSSGCN	83.82	76.80	78.96	75.77	75.04	74.20

Table 3. Results of ablation experiments.

Models	Res14		Laptop		Twitter	
	Acc	F1	Acc	F1	Acc	F1
w/o-A_s	82.48	74.64	77.68	73.96	73.56	71.91
w/o-A_a	82.89	74.90	77.23	74.26	73.41	71.93
w/o-D_t	83.02	75.32	77.85	73.84	74.15	72.51
CSSGCN	83.82	76.80	78.86	75.77	75.04	74.20

semantic relationship between aspectual words and context is indispensable to the model. Removing the components with syntactic structure information leads to a decrease in the accuracy of Res14, Laptop, and Twitter by 0.80%, 1.11%, and 0.89%, respectively, which proves that the addition of syntactic structure information to the model improves the categorization accuracy, so that each component contributes to the model's ability to learn richer information and thus improve the model's performance.

4.4. Effect of GCN Layers on Results

In order to study the effect of the number of GCN layers on the CSSGCN model, this paper conducts experiments on three public datasets to analyze the effect of the accuracy and F1 value when the number of GCN layers L is used. The experimental results are shown in Figure 4 and Figure 5, and the model works best when the number of GCN layers L = 2. When the number of GCN layers is less than 2, the node information cannot be transferred and updated, and the semantic information between the nodes is not sufficiently learned. If the number of GCN layers is too many, the parameters will be increased accordingly which will cause the model to be difficult to converge or the gradient will disappear.

4.5. Example Analysis

In order to verify whether the fused syntactic and semantic information is helpful to the model, this paper selects an instance on Res14 for the visual analysis of the weights, as shown in Figures 6-8, where the darker the color means the larger the value of the weights. The sample contains the aspectual word "staff", from a syntactic and semantic point of view, if we want to correctly recognize the sentiment polarity of the aspectual counterpart, we should pay more attention to

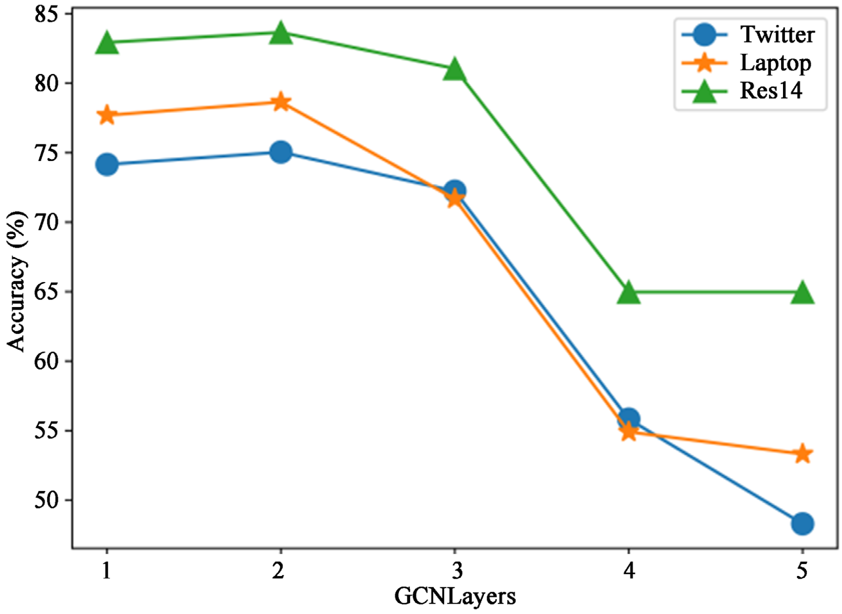

Figure 4. Effect of the number of GCN layers on accuracy.

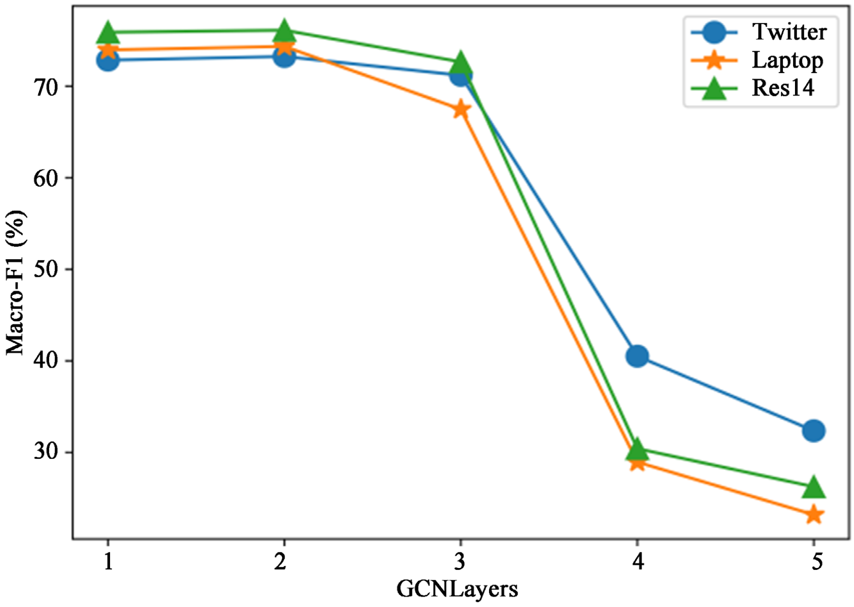

Figure 5. Effect of F1 values of GCN layers.

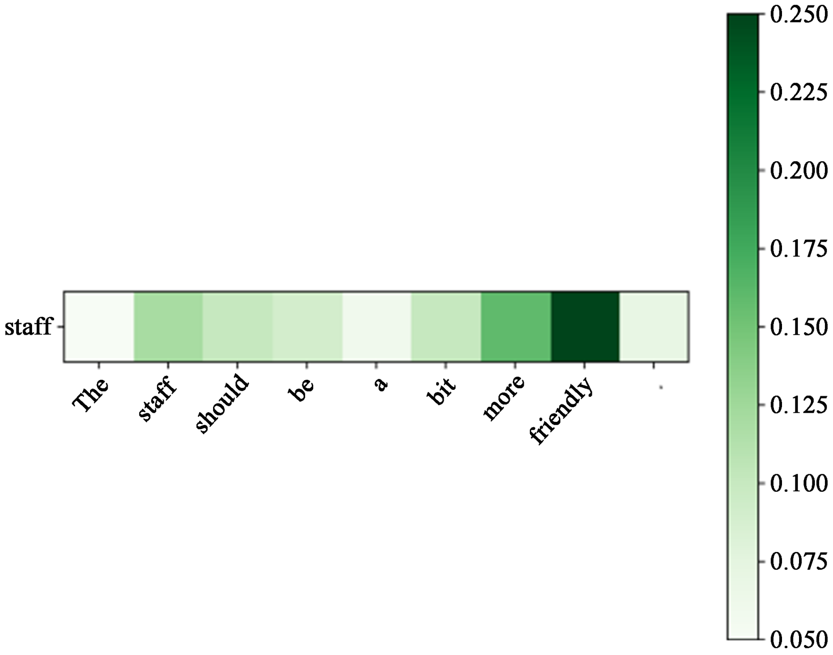

Figure 6. Weight of self-attention.

"should be", but the self-attention mechanism incorrectly focuses on "dreadful". The self-attention mechanism incorrectly focuses on "dreadful", but for the aspectual attention mechanism, although it notices "should be", the main focus is still on "dreadful". For the aspectual attention mechanism, although it notices "should be", the main focus is still on "dreadful", and if the model understands the sentence in this way, it will make wrong predictions. For the model that fuses syntactic and semantic information, the main focus of attention is on "should be", which indicates that fusing syntactic and semantic information facilitates

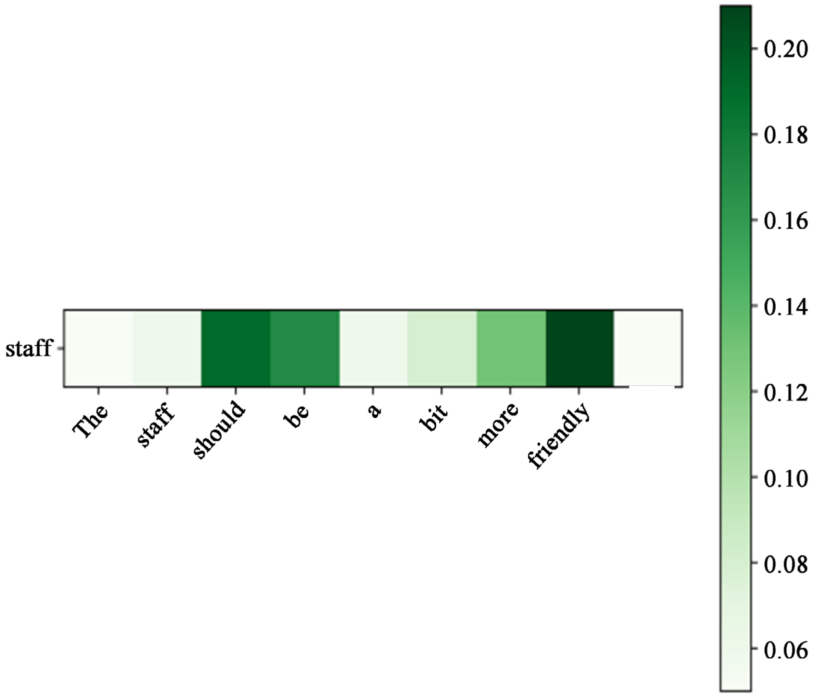

Figure 7. Weight of aspect-attention.

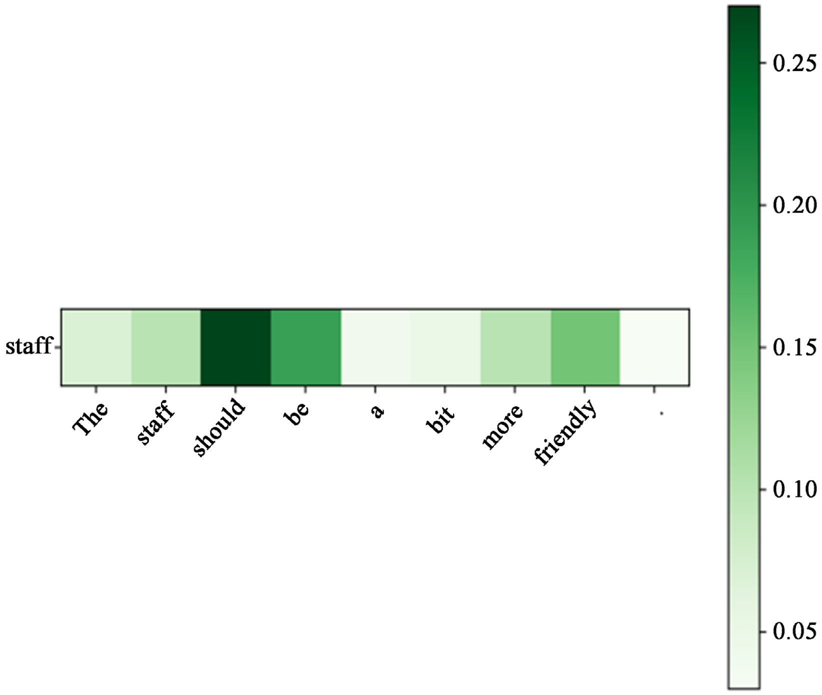

Figure 8. Weight of the fused information.

the model to judge the correct sentiment polarity.

5. Conclusion

In the aspect-level sentiment analysis task, this paper proposes a CSSGCN model that fuses semantic information with syntactic structure information, which solves the problem that semantic and syntactic information cannot be effectively fused. Two attention mechanisms are utilized to fully obtain the semantic information in the utterance, and the dependency tree is utilized to extract the structural information of the utterance. The experimental results as well as the example analysis show that CSSGCN can effectively improve the classification accuracy by using the fused information.

Acknowledgements

This work was supported by the National Social Science Foundation Project "Research on Intelligent Intelligence Perception Driven by Multi-Source Data Fusion", China (Item No. 21BTQ071).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Wang, L., Ma, H.W. and Lyu, H.H. (2022) A Review of Aspect Level Text Sentiment Analysis. *Journal of Computer Applications*, **42**, 1-9.

[2] Tan, C.P. (2022) Fine-Grained Text Sentiment Analysis Research Review. *Journal of University Library*, **40**, 85-99+119.

[3] Nazir, A., Rao, Y., Wu, L.W., *et al.* (2020) Issues and Challenges of Aspect-Based Sentiment Analysis: A Comprehensive Survey. *IEEE Transactions on Affective Computing*, **99**, 1.

[4] Zhang, C., Li, Q.C. and Song, D.W. (2019) Aspect-Based Sentiment Classification with Aspect-Specific Graph Convolutional Networks. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, November 2019, 4568-4578. https://doi.org/10.18653/v1/D19-1464

[5] Sun, K., Zhang, R.C., Samuel, M., *et al.* (2019) Aspect-Level Sentiment Analysis via Convolution over Dependency Tree. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, November 2019, 5679-5688. https://doi.org/10.18653/v1/D19-1569

[6] Zhang, M. and Qian, T.Y. (2020) Convolution over Hierarchical Syntactic and Lexical Graphs for Aspect Level Sentiment analysis. *Proceedings of the* 2020 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), November 2020, 3540-3549. https://doi.org/10.18653/v1/2020.emnlp-main.286

[7] Xu, G.T., Liu, P.Y., Zhu, Z.F., *et al.* (2021) Attention-Enhanced Graph Convolutional Networks for Aspect-Based Sentiment Classification with Multi-Head Attention. *Applied Sciences*, **11**, Article 3640. https://doi.org/10.3390/app11083640

[8] Wang, R.Y., Tao, Z.Y., Zhao, R.J., *et al.* (2022) Multi-Interaction Graph Convolutional Networks for Aspect-Level Sentiment Analysis. *Journal of Electronics and Information Technology*, **44**, 1111-1118.

[9] Cui, S.G., Chen, S.Q. and Du, X. (2023) Dual Graph Attention Network Model for

Target Emotion Analysis. *Journal of Xidian University*, **50**, 12.

[10] Zhu, L., Zhu, X.F., Guo, J.F., *et al.* (2022) Exploring Rich Structure Information for Aspect-Based Sentiment Classification. *Journal of Intelligent Information Systems*, **60**, 97-117. https://doi.org/10.1007/s10844-022-00729-1

[11] Zhuang, C.Y. and Qiang, M. (2018) Dual Graph Convolutional Networks for Graph-Based Semi-Supervised Classification. *The* 2018 *World Wide Web Conference*, April 2018, 499-508. https://doi.org/10.1145/3178876.3186116

[12] Pennington, J., Socher, R. and Manning, C.D. (2014) Glove: Global Vectors for Word Representation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing*, Doha, October 2014, 1532-1543. https://doi.org/10.3115/v1/D14-1162

[13] Fan, F.F., Feng, Y.S. and Zhao, D.Y. (2018) Multi-Grained Attention Network for Aspect-Level Sentiment Classification. *Proceedings of the* 2018 *Conference on Empirical Methods in Natural Language Processing*, Brussels, October-November 2018, 3433-3442. https://doi.org/10.18653/v1/D18-1380

[14] Maria, P., Dimitris, G., John, P., Androutsopoulos, I. and Manandhar, S. (2014) SemEval-2014 Task 4: Aspect Based Sentiment Analysis. *Proceedings of the* 8*th International Workshop on Semantic Evaluation* (*SemEval* 2014), Dublin, August 2014, 27-35.

[15] Dong, L., Wei, F.R., Tan, C.Q., *et al.* (2014) Adaptive Recursive Neural Network for Target-Dependent Twitter Sentiment Classification. *Proceedings of the* 52*nd Annual Meeting of the Association for Computational Linguistics* (*Volume* 2*: Short Papers*), Baltimore, June 2014, 49-54. https://doi.org/10.3115/v1/P14-2009

[16] Wang, Y.Q., Huang, M.L., Zhu, X.Y., *et al.* (2016) Attention-Based LSTM for Aspect-Level Sentiment Classification. *Proceedings of the* 2016 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Austin, November 2016, 606-615. https://doi.org/10.18653/v1/D16-1058