

Article

MSGAT-Based Sentiment Analysis for E-Commerce

Tingyao Jiang ¹, Wei Sun ^{1,*}  and Min Wang ²¹ College of Computer and Information, China Three Gorges University, Yichang 443002, China² Hubei Three Gorges Polytechnic, Yichang 443002, China

* Correspondence: 202108120021021@ctgu.edu.cn

Abstract: Sentence-level sentiment analysis, as a research direction in natural language processing, has been widely used in various fields. In order to address the problem that syntactic features were neglected in previous studies on sentence-level sentiment analysis, a multiscale graph attention network (MSGAT) sentiment analysis model based on dependent syntax is proposed. The model adopts RoBERTa_WWM as the text encoding layer, generates graphs on the basis of syntactic dependency trees, and obtains sentence sentiment features at different scales for text classification through multilevel graph attention network. Compared with the existing mainstream text sentiment analysis models, the proposed model achieves better performance on both a hotel review dataset and a takeaway review dataset, with 94.8% and 93.7% accuracy and 96.2% and 90.4% F1 score, respectively. The results demonstrate the superiority and effectiveness of the model in Chinese sentence sentiment analysis.

Keywords: sentiment analysis; graph attention networks; dependent syntactic analysis; NLP



Citation: Jiang, T.; Sun, W.; Wang, M. MSGAT-Based Sentiment Analysis for E-Commerce. *Information* **2023**, *14*, 416. <https://doi.org/10.3390/info14070416>

Academic Editor: Katsuhide Fujita

Received: 1 June 2023

Revised: 4 July 2023

Accepted: 14 July 2023

Published: 19 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sentiment analysis involves the task of analyzing the sentiment of given textual data, and, with the rapid development of Internet applications in recent years, the demand for this task has been increasing, making it a hot research topic. Currently, most of the textual data on the Internet exist in the form of short comments; thus, it is important to study sentiment analysis at the sentence level. Sentiment analysis at the sentence level has two main subtasks: vector representation of the input text and sentiment feature extraction.

The current mainstream feature representation of text uses the word embedding technique [1,2], which represents the input text as a multidimensional vector. Before the birth of neural networks, tf-idf was used to represent text. The text representation was obtained by calculating the word frequency of each word and dividing it by the total number of documents for normalization. The word vector representation obtained by this method greatly depends on the given corpus documents, and there are problems such as an inability to learn contextual information and distribution features; hence, the text representation obtained is less effective. Furthermore, the feature representation obtained using this method is similar to one-hot coding, and the problem of curse dimensionality is inevitable as the number of words in the corpus increases.

In order to solve the problems of curse dimensionality and semantic gaps, subsequent studies focused on the construction of distributed low-dimensional dense word vectors. On the basis of this idea, the word2vec model [3], which can use two different models (CBOW and Skip-Gram), was proposed. Since the implementation of word2vec is based on the sliding window approach, only local features can be extracted, and more information cannot be effectively fused. To solve the problem of word2vec, the GloVe model [4] was proposed to combine the global information of the corpus.

However, both word2vec and GloVe have the problem that the representation of word vectors is the same in different contexts, which obviously does not meet the requirements of natural language. For example, the word apple obviously has different meanings

with regard to an Apple phone and an apple orchard, and the representation of both appearing in the corresponding utterance should not be the same; therefore, in order to incorporate contextual information, pretrained language models such as ELMo [5] and BERT [6] have emerged. Currently, the BERT family of models performs better in all major natural language processing tasks. In this paper, a variant of BERT, the RoBERTa_WWM model [7,8], is used as the text encoding layer.

The existing methods for sentiment feature extraction mainly include the convolutional neural network (CNN), the recurrent neural network (RNN) [9] and its variants, and the graph neural network (GNN). Kim [10] proposed a multichannel CNN to extract text sentiment features and obtained good results. However, the CNN model uses convolutional kernels to extract local features from the input data, ignoring the whole of the sentence, and the obtained features lack global contextual information.

RNN, as a kind of temporal network, is more consistent with the sequence structure characteristics of the sentence text compared with CNN, and it can better extract the global features of the sequence. Since the structure of RNN itself is passed backward in a time sequence, there is a problem in that the gradient of the sequence disappears during the passing process. To solve this problem, a series of RNN variants have been proposed, among which the most widely used ones are the long short-term memory model (LSTM) [11] and the gated recurrent unit model (GRU).

In recent years, graph neural networks (GNNs) have been applied to several natural language processing tasks. A GNN can update the node features in the network according to the edges in the network and obtain the global information of the whole graph by aggregating the nodes in the graph. Although both GNN and RNN acquire global features of sentences, RNN only learns the overall features of a sentence serially from end to end by treating the input sentence as a sequence, and this approach ignores the syntactic structure of the utterance. In contrast, natural language is based on a certain syntactic foundation. The graph network-based model proposed in this paper constructs a syntactic dependency tree of sentences through syntactic relations, obtains a graph corresponding to the tree, and learns the corresponding global features of sentences by updating the edges in the graph. The main contributions of this paper are as follows:

1. A multiscale GAT [12] model based on syntactic dependency trees is proposed to extract sentiment features of sentences as a function of their syntactic structure, which solves the problem of CNN, LSTM, and other models ignoring the syntactic features of sentences.
2. A method for constructing graph networks after encoding Chinese utterances is proposed. The representation of node features in the syntactic spanning tree is obtained when syntactic analysis is performed on Chinese utterances.
3. Two Chinese e-commerce review datasets are constructed; the proposed model is applied to the datasets, and good results are obtained.

2. Related Work

Sentiment analysis tasks are currently classified into sentiment lexicon-based algorithms, machine learning-based algorithms, and deep learning-based algorithms [13,14]. Sentiment lexicon-based algorithms require domain personnel to create a specialized sentiment lexicon; then, the text to be classified is divided into words, the words in the text are sentiment labeled using the constructed sentiment lexicon, and the sentiment score is calculated using semantic rules to derive sentiment tendency. Hu et al. [15] proposed that the key to determining sentiment polarity is the degree of sentiment of adjectives, and they created a sentiment lexicon for sentiment analysis. Yang et al. [16] proposed a method to construct domain-specific lexicons. Zhou et al. [17] proposed a method to construct a sentiment dictionary for Chinese microblogs. The main core of the sentiment lexicon-based approach lies in the construction of the sentiment lexicon, the suitability of the approach greatly depends on the degree of the construction of the lexicon, and the construction

process of the lexicon itself is complicated and mostly domain-specific, making it less applicable.

Machine learning-based methods are essentially feature engineering extraction methods, i.e., extracting different classes of features from a labeled dataset. Pang et al. [18] first used machine learning for sentiment analysis by comparing SVM, NB, and ME with multiple feature combinations for the classification of movie reviews, and they concluded that the combination of monadic features and SVM worked best. Wikarsa et al. [19] used a plain Bayesian classifier for Twitter users' comment sentiment analysis, classifying emotions into six categories, with an experimental accuracy of 83%. Ying Su et al. [20] proposed an unsupervised model for text sentiment analysis by combining a plain Bayesian model and a latent Dirichlet distribution, which outperformed other unsupervised models in terms of correctness. The machine learning-based approach for sentiment analysis may have problems such as sparse feature vectors, dimensional explosion, and difficult feature extraction when feature extraction is performed.

Deep learning is a multilayer representation learning algorithm with a deeper network structure and stronger expressive power compared to traditional machine learning algorithms. Sentiment analysis based on deep learning generally takes the word vector obtained from word embedding training as the input of the sentiment analysis model, and then performs feature extraction through the feature extraction network of the model, after which the sentiment polarity is obtained by means of classification. Yuan [21] combined LSTM and Word2Vec for sentiment analysis of book reviews. Wang et al. [22] used LSTM for Twitter sentiment analysis and obtained good results. Wang et al. [23] combined CNN and LSTM models, using CNN to deal with local dependencies and LSTM to deal with remote dependencies, and achieved better results. Feng et al. [24] proposed a sentiment analysis scheme based on a convolutional neural network and attention mechanism for a sentiment analysis scheme and achieved good results. Li et al. [25] used CNN and BiLSTM for feature fusion and obtained better results than a single-feature network.

Among several methods currently applied to sentiment analysis, deep learning-based methods outperform the others in terms of the complexity of work, the ease of implementation, and the final experimental results.

3. MSGAT

3.1. Model Architecture

As shown in Figure 1, MSGAT consists of seven modules: input, sentence encoding, construct parse tree, vertex representation, generate edge list, feature extraction, and classification. Being an end-to-end model, it accepts sentences as the input and outputs sentiment polarity.

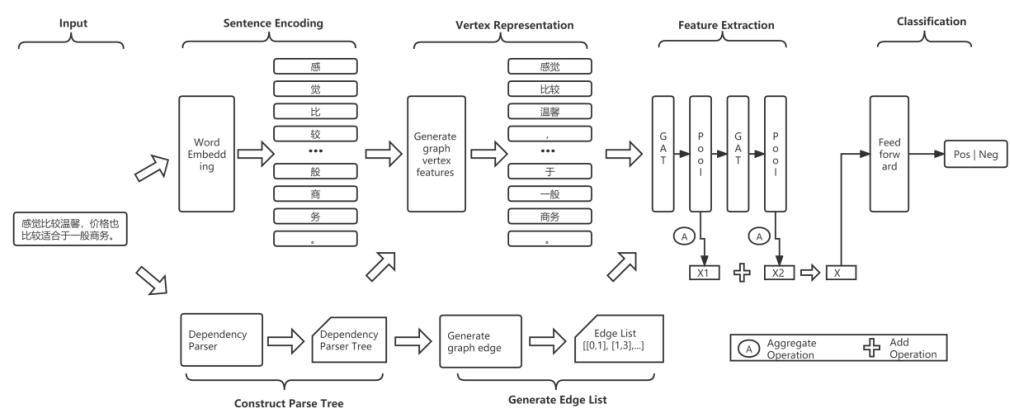


Figure 1. Model structure. The sentences and phrases in the figure are in Chinese, which are obtained from the hotel review dataset.

A preprocessed dataset is used as the model input; a detailed description of the dataset is given in Section 4. For a given input sentence, each word in the Chinese sentence is first represented using a vector, which is generated by the sentence encoding module. Because the feature of the input sentence is calculated via GAT, vertices and edges are needed. The vertex representation module generates vertices, while edges are output by the generate edge list module. Both of these modules require the construct parse tree module’s output as the input. After obtaining vertices and edges, the feature extraction module uses GAT to output the sentiment feature, which is the input of the classification module.

For vertex representation, due to the nature of Chinese phrases, a single word cannot normally be a phrase, and dependency parsing of Chinese sentences is based on phrase components. It is necessary to aggregate the vector generated by the sentence encoding module. In the phrase in the figure, which consists of two words, the feature representation is obtained by aggregating the two vectors.

The feature extraction module includes two components: GAT and the pooling layer. Through the operation of GAT and pooling, the vertex features in each graph network are updated. By aggregating the vertex features, the sentiment features corresponding to the sentence at different scales are obtained. The features at different scales are fused as the final feature vector.

3.1.1. Sentence Encoding

The word embedding adopts the RoBERTa_WWM pretraining model. The RoBERTa model is an improved BERT model, mainly modifying the pretraining method, through dynamic masking, elimination of the next sentence prediction task, expansion of the batch size, etc. For WWM (whole word mask), for example, when predicting the sentence “There is an apple tree.”, the original mask will pick the word “apple” as the mask part for prediction, while WWM will mask the words “apple tree” as the whole word mask, which is more reasonable.

Given a sentence S of length N , the vector representation of the sentence $z = \{z_1, z_2, \dots, z_N\}$ is obtained by encoding it with the RoBERTa model. For each z_i , a 768-dimensional vector is represented.

3.1.2. Construct Parse Tree

Natural languages have their own syntactic structures. A specific sentence is constructed on the basis of dependency syntactic relations according to the meaning to be expressed. The dependency relations, here referred to as parse trees, can be obtained by performing a dependency syntactic analysis of the sentence.

For example, a sample sentence entered into Figure 1. After the dependency syntactic analysis, a parse tree is obtained as shown in Figure 2.

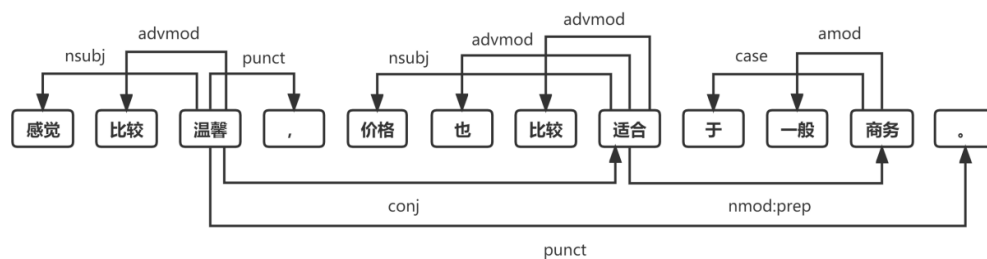


Figure 2. The dependency syntactic tree structure of the sentence. The sentence in Chinese is a sample of the hotel review dataset.

The relationships of the sentence shown in the diagram are represented in Table 1.

Table 1. Dependency relationship table.

Relation	Description
advmod	Adverbial modifier
punct	Punctuation
case	Dependencies
amod	Adjective modifier
conj	Parallelism
nmod	Compound noun modifier
nsubj	Subject–predicate relationship

3.1.3. Vertex Representation

According to the parse tree of the sentence, the word vectors are aggregated to obtain the feature representation of each node in the tree. For Chinese, the node often refers to phrase, which consists of several words. Thus, it is not necessary to aggregate vectors if the node is a single word.

For node i in the tree, assuming that it consists of the words $[z_k, z_{k+1}]$ corresponding to sentence, the feature representation of node i is as in Equation (1).

$$T_i = AGG(z_k, z_{k+1}), \quad (1)$$

where T_i denotes the feature of node i . AGG is the aggregation operation, which can be selected as the summation, mean, maximum, etc. One can also use concat to stitch the features as the node's features. Here, we choose mean aggregation, which is simple to implement and works well.

3.1.4. Generate Edge List

The edges of nodes are constructed according to the parse tree, and the set of edges is used as the updated adjacency matrix, which is input into GAT. As the tree structure shows in Figure 2, there is a relational edge between feel and warm; assuming that these two nodes are numbered 1 and 2, the generated edge is denoted as (1,2). By traversing the whole tree, the set of all edges is obtained, denoted as A . A is the adjacency matrix of the sentence. When each node is updated, the neighboring vectors are added to the node itself by the adjacency matrix according to the attention weights. For the vector of the node itself, it also needs to be added when updated; hence, the edges of all nodes pointing to themselves are added to the set.

In summary, the obtained A is expressed in Equation (2).

$$A = \begin{cases} (i, j) & i \neq j; i, j \in E \\ (i, i) & i \in E' \end{cases} \quad (2)$$

where E is the set consisting of all nodes in the tree, and i and j are the nodes in the set.

3.1.5. Feature Extraction

After obtaining the node features and adjacency matrix, it is input into the GAT for node feature updating, followed by the pooling operation for down-sampling, and then the sentence features are obtained at the current scale by aggregating the overall node features. After two rounds of feature learning in the network, the sentence features at two different scales are obtained. These features are fused as the final vector of the sentence, which is the sentiment feature vector of the corresponding sentence.

The GAT takes the feature vectors of the nodes and the adjacency matrix of the nodes as the input, and updates features of the nodes through the adjacency matrix via the attention mechanism. Suppose that the input node features $T = \{T_1, T_2, \dots, T_N; T_i \in R^F\}$, where N denotes the number of nodes, and F denotes the dimension of features. In order to obtain sufficient expressiveness, set a weight matrix $W \in R^{F' \times F}$, and perform a linear

transformation on T_i to obtain WT_i . Then, perform the self-attention and calculate the attention coefficients using Equation (3).

$$e_{ij} = a(WT_i, WT_j), \tag{3}$$

where j is a node adjacent to i ; $a \in R^{2F'}$ is used to map the spliced high-dimensional features into a concrete real number as the attention of node i to node j .

Afterward, the normalization operation is carried out for the attention coefficient as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(e_{ik}))}. \tag{4}$$

After obtaining the attention coefficients, feature fusion is performed on the original features as the output features for each node as follows:

$$T'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} WT_j \right). \tag{5}$$

TopKPooling [26] is used as the pooling layer. The filtering ratio is set as $0 < \text{ratio} < 1$, denoted as r ; then, the k highest weighted nodes from the graph nodes are filtered to form a new graph, and the adjacency matrix is updated to the matrix of selected nodes. Thus, $k = rN$, where N is the number of nodes in the original graph. Setting a learnable matrix $P \in R^{F' \times 1}$, the weights are calculated according to Equation (6).

$$y = \frac{T'_i P}{\|P\|}. \tag{6}$$

Then, k nodes are filtered as in Equation (7).

$$i = \text{top}_k(y). \tag{7}$$

The selected nodes are updated with features according to the weight values, as in Equation (8).

$$T'' = (T' \tanh(y))_i. \tag{8}$$

The adjacency matrix is updated as in Equation (9).

$$A' = A_{i,i}. \tag{9}$$

Extracting the global features of the new graph obtained after pooling can be achieved in three ways: global average pooling, global maximum pooling, and global summation. The global average pooling is used as the extraction method, as in Equation (10).

$$x = \frac{1}{N_i} \sum_{n=1}^{N_i} T''_n. \tag{10}$$

The global features of three different scales are (x_1, x_2, x_3) , summed as the final global feature x_s .

3.1.6. Classification

The output of the feature extraction is fed into the feedforward neural network to obtain the final sentiment polarity. A softmax function is used as the activation function, defining a positive label of 1 and a negative label of 0.

The cross-entropy loss function was chosen to calculate the loss with Equation (11).

$$L = \frac{1}{N} \sum_i -[y_i \log(y_i) + (1 - y_i) \log(1 - y_i)], \tag{11}$$

where y_i is the true label, and y'_i is the network-predicted value.

4. Experiment and Discussion

4.1. Dataset

The experiments are based on two Chinese datasets, a hotel review dataset and a takeaway review dataset. The data distribution for these datasets is shown in Table 2.

Table 2. Dataset distribution.

Dataset	Positive Emotions		Negative Emotions		Total
	Train	Test	Train	Test	
Hotel review dataset	3726	1596	1711	732	7766
Takeaway review dataset	2800	1200	5592	2394	11,986

The hotel review dataset was collected from the Meituan platform, which is an E-commerce platform that is well known in China. The takeaway review dataset was collected from the ELEME platform, which belongs to Alibaba. These comments were obtained from the comments section of the corresponding platform, which fully reflects the authenticity and validity of the data. All of the comments were written by Chinese people in Mandarin, and they were collected at the beginning of this year.

As is customary in China, people usually write reviews out of appreciation for good service and complaints about unsatisfactory service. Therefore, neutral reviews are rare. For this reason, we removed these comments and retained only the positive and negative ones, and this work was carried out simultaneously when annotating the dataset. Invalid comment data also needed to be removed. After these processes, we obtained the amount of data shown in Table 2.

Of course, these data are not the final input data. The data in the dataset were text-preprocessed to remove emoticons and invalid characters from the comments; the format of the processed data is shown in Table 3.

Table 3. Data format (for readability, the original text has been translated to English).

Dataset	Content
Hotel review dataset	(1, Business king room, the room is large, the bed is 2 m wide, the overall feeling of economy is good!) (0, I booked a suite in the secondary floor during the National Day, and it was more than a little worse, the furniture was shabby and the TV was incredibly small and unimaginably spartan.)
Takeaway review dataset	(1, Delicious! Fast! The packaging has quality too . . . restaurant food without leaving home!) (0, Too bad. I waited 2 h for the beef, and I was about to throw up; never again.)

In the table, 1 indicates a positive sentiment label and 0 indicates a negative sentiment label. The tags and comment data are separated by commas.

4.2. Experimental Environment

The hardware devices and software versions used for the experiments are listed in Table 4.

Table 4. List of experimental environments.

Classification	Specific Description
Hardware type	CPU: Intel(R) Xeon(R) W-2255 CPU @ 3.70 GHz GPU: NVIDIA GeForce RTX 3080 Ti Memory: 64 GB Hard disk: 4 TB OS: Windows 10
Software version	python: 3.9.12 torch: 1.12.0 pycharm: 2022.1.3

4.3. Baselines

The BERT model has excellent performance in all major subtasks in NLP, and the pretrained model can be directly used in downstream tasks. The BERT + BiLSTM model uses the BERT model as the text encoding layer, and then BiLSTM extracts features containing textual context for sentiment classification; the TextCNN-based model uses multiple convolutional kernels of different sizes for feature extraction and splices multiple features for sentiment classification.

In order to avoid the influence of the coding layer on the final results, the above models all use the RoBERTa model as the text coding layer. In terms of training parameters, the same batch size, learning rate, and epoch were set as stated in Table 5.

Table 5. Parameter settings.

Parameter Name	Description	Value
Batch size	Volume of data per batch	32
Epoch	Number of times the dataset was learned	20
Learning rate	Learning rate	10^{-5}
Optimizer	Optimizer	AdamW
Dropout	Random drop rate	0.5

4.4. Experimental Results and Analysis

Experiments were conducted for each model, and the results obtained are shown in Table 6.

Table 6. Experimental results.

Model	Hotel Review Dataset		Takeaway Review Dataset	
	Accuracy	F1	Accuracy	F1
RoBERTa	90.84	93.46	90.76	85.29
RoBERTa + BiLSTM	93.49	95.27	91.99	87.73
RoBERTa + TextCNN	93.62	95.31	92.51	88.62
RoBERTa + MSGAT	94.79	96.23	93.72	90.44

As can be seen from the data in the table, RoBERTa + MSGAT performed better on both datasets and achieved good scores in terms of both accuracy and F1 scores. TextCNN and BiLSTM were similar in terms of performance, and the difference between them varied depending on the initial values of different weights, but the difference was essentially nonsignificant. The results obtained using only the RoBERTa model were not as good as the other three models, but they also achieved good scores, verifying the strong performance of the RoBERTa model itself.

For a more visual observation of the convergence of each model, the loss of each model on the dataset is shown in Figure 3.

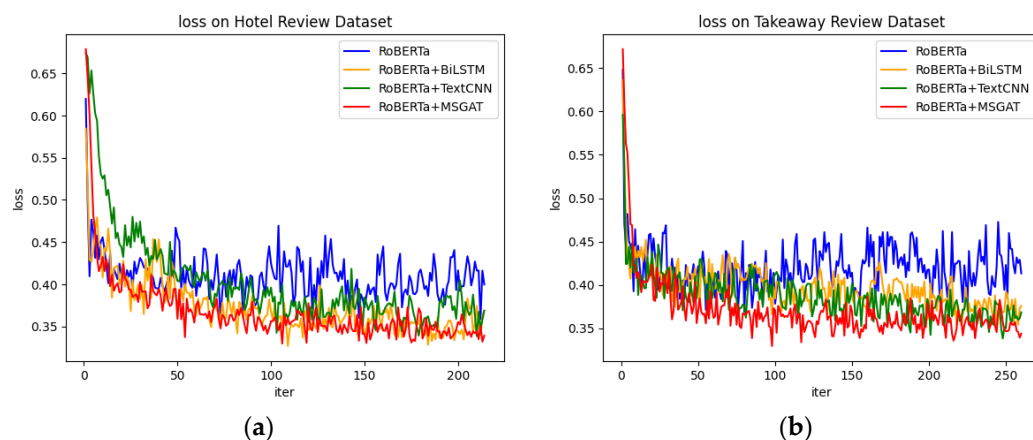


Figure 3. Convergence of each model. (a) Loss convergence on the hotel review dataset; (b) Loss convergence on the takeaway review dataset.

It is obvious from Figure 3 that the convergence of the models was similar except for the RoBERTa model, whereby the convergence of the RoBERTa + MSGAT model was a little more stable. Since the RoBERTa model reached convergence in the early stage of training, after setting the same epoch as the other models, the model no longer converged subsequently, but varied within a certain range.

4.5. Ablation Study

The ablation experiments were conducted to determine the effects of the number of graph attention network layers and different graph network feature aggregation methods on the experimental results. The hotel review dataset was chosen here as the comparison experimental dataset because it had relatively fewer data for the model to learn, more clearly reflecting the difference in results between different parameters.

Table 7 shows the experimental results with different numbers of layers of the GAT.

Table 7. Experimental results with different numbers of the layers.

Number of GAT Layers	Accuracy	F1
Single GAT	94.01	95.63
Double GAT	94.79	96.23
Triple GAT	94.36	95.88

From the data in the table, it can be intuitively seen that the best results were obtained using two-layer GAT, while the three-layer GAT was slightly worse than the two-layer GAT, but better than one-layer GAT. It can be considered that the one-layer GAT network was not sufficient for feature extraction, which made the classification results poor, and the three-layer GAT was over-fused, which made the features redundant and affected the classification effect.

Table 8 shows the experimental results using different feature aggregation methods.

Table 8. Experimental results of different polymerization methods.

Global Aggregate Function	Accuracy	F1
global_mean_pool	94.79	96.23
global_max_pool	94.45	95.96

From the results in the table, it can be seen that the above two aggregation methods did not have a great impact on the experimental results, and both could provide very good results. There is no definite conclusion as to which of the two commonly used aggregation methods, namely, mean aggregation and maximum aggregation, was more effective. For the dataset in this paper, the effects of the two methods were relatively similar, and the effect of mean aggregation was relatively better.

5. Conclusions

In Chinese sentence-level sentiment analysis, we propose using an MSGAT to combine the syntactic structure and the sentiment features of sentences as a whole. This solves the problem that traditional deep learning methods, such as the long short-term memory model (LSTM) and the convolutional neural network (CNN), only extract serialized and local sentiment features when extracting sentiment features of sentences, ignoring the syntactic structure information. Through the feature fusion method, the transformation of input sentences into syntactic dependency trees is achieved. It is experimentally demonstrated that MSGAT achieved better performance than BiLSTM and TextCNN on the proposed Chinese datasets. The performance on larger datasets is yet to be verified. For sentence sentiment polarity analysis as a dichotomous classification, the application of sentiment multiclassification tasks needs to be further considered as a research direction.

Author Contributions: Conceptualization, T.J., W.S. and M.W.; data curation, T.J., W.S. and M.W.; methodology, T.J. and W.S.; writing—original draft, T.J. and W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61871258.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hinton, G.E. Learning distributed representations of concepts. In Proceedings of the Eighth Conference of the Cognitive Science Society, Amherst, MA, USA, 15–17 August 1986; pp. 1–12.
2. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2006**, *3*, 1137–1155.
3. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
4. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 26–28 October 2014.
5. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. *arXiv* **2018**, arXiv:1802.05365.
6. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
7. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
8. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting Pre-Trained Models for Chinese Natural Language Processing. *arXiv* **2020**, arXiv:2004.13922.
9. Mikolov, T.; Karaflav, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the INTERSPEECH 2010, Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
10. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882.
11. Sundermeyer, M.; Schluter, R.; Ney, H. LSTM Neural Networks for Language Modeling. In Proceedings of the INTERSPEECH, Portland, OR, USA, 9–13 September 2012; pp. 194–197.
12. Velickovi, P.; Cucurull, G.; Casanova, A.; Liò, P.; Romero, A. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
13. Wang, T.; Yang, W. A review of text sentiment analysis methods. *Comput. Eng. Appl.* **2021**, *57*, 11–24.
14. Zhou, C.; Li, S.; Xu, Y. Research on text sentiment analysis. In Proceedings of the China Computer Users Association Web Application Branch 2018 22nd Annual Conference on New Technologies and Applications for the Web, Suzhou, China, 8–10 November 2018; pp. 302–305.

15. Hu, M.; Bing, L. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177.
16. Yang, M.; Peng, B.; Chen, Z. A Topic Model for Building Fine-Grained Domain-Specific Emotion Lexicon. In Proceedings of the Association for Computational Linguistics (ACL), Baltimore, MD, USA, 22–27 June 2014; pp. 421–426.
17. Zhou, Y.; Yang, A.; Lin, J. A method for constructing Chinese microblogging emotion dictionary. *J. Shandong Univ.* **2014**, *44*, 36–40.
18. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up: Sentiment Classification Using Machine Learning Techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, USA, 6 July 2002; pp. 79–86.
19. Wikarsa, L.; Thahir, S.N. A Text Mining Application of Emotion Classifications of Twitter’s Users Using Naive Bayes Method. In Proceedings of the Wireless and Telematics (ICWT), Manado, Indonesia, 17–18 November 2015.
20. Su, Y.; Zhang, Y.; Hu, P.; Tu, X.H. Sentiment analysis based on the combination of plain Bayesian and latent Dirichlet distributions. *Comput. Appl.* **2016**, *36*, 1613–1618.
21. Chai, Y. Research on sentiment analysis of book review text based on LSTM and Word2vec. *Inf. Technol.* **2022**, *7*, 59–64.
22. Wang, X.; Liu, Y.; Sun, C.; Wang, B.; Wang, X. Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Beijing, China, 26–31 July 2015.
23. Wang, J.; Yu, L.-C.; Lai, R.K.; Zhang, X. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, 7–12 August 2016.
24. Feng, X.J.; Zhang, Z.W.; Shi, J.C. Text sentiment analysis based on convolutional neural network and attention model. *Comput. Appl. Res.* **2018**, *35*, 1434–1436.
25. Li, Y.; Dong, H.-B. Text sentiment analysis based on CNN and BiLSTM network feature fusion. *Comput. Appl.* **2018**, *38*, 3075–3080.
26. Gao, H.; Ji, S. Graph U-Nets. *Inst. Electr. Electron. Eng. (IEEE)* **2022**, *44*, 4948–4960. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.