Applied Intelligence (2023) 53:2088–2097
https://doi.org/10.1007/s10489-022-03526-z

Exploring semantic awareness via graph representation for text classification

Yahui Li[1] · Yifan Liu[1] · Zhenfang Zhu[1] · Peiyu Liu[1]

Accepted: 17 March 2022 / Published online: 5 May 2022

Abstract

Text classification is a fundamental problem in natural language processing. Nowadays, text classification based on GNN attracts the attention of researchers. However, the existing works not fulfill well the transmission of contextual semantic information, and they pay more attention to capturing the local features instead of global. Such methods ignore the importance of keyword information features, so they can not fully mine the text-level semantic representation. To relieve such problems, we propose the GText model for discovering the basic features with words and establishing a deeper relationship representation between words and documents. Specially, we utilize semantic features graphs to achieve text semantic representation. Meanwhile, we propose semantic information passing(SIP) mechanism to transmit contextual semantic information, which can enhance the semantic representation from multi-views. In addition, the gate mechanism can further mine the explicit keywords of the whole document. With GText, the test accuracy on MR improved about 2% and on Ohsumed at most 9%, which illustrates GText can better achieve the mining and transmission of text semantic information. Experiments on several authoritative datasets show that our method is superior to the existing text classification methods.

Keywords Text classification · Graph neural network · Semantic information

1 Introduction

As a primary task in the NLP field, text classification is a foundation for many NLP tasks, and it has received continuous endeavors from researchers due to its wide spectrum of applications, such as sentiment analysis [1], topic labeling [2], and disease diagnoses [3]. In the early text classification tasks, statistical models are dominant, such as Naive Bayes (NB) [4], k-nearest neighbor (KNN) [5], and support vector machine (SVM) [6]. These traditional methods use Bag-of-Words(BoW) sparse representation of texts, which contributes to designing models for binary, multi-class, and multi-label classification problems. Although traditional text classification methods could reach reasonable performance, these methods still have some issues, including sparse features, the representation ability is limited, and so forth. Therefore, these methods can not make a fully semantic understanding and produce considerable features to represent the natural language.

With the development of deep learning, these problems are gradually relieved. Deep learning is to learn a set of nonlinear transformations, then integrate feature engineering directly into the output, and finally integrate feature engineering into the process of model fitting. For instance, Convolutional Neural Network(CNN) [7] and Recurrent Neural Network(RNN) [8], they are all essential text classification methods. Based on them, some extended models appear. For example, TextCNN [7],TextRNN [9], TextRCNN [10], fastText [11], long short-term memory(LSTM) and Bi-LSTM. Compared with traditional models, these models have superior performance. The key is that deep learning methods have better representation for texts, which helps to get significantly improved performance using even off-shelf linear classifier models.

However, these methods all have many drawbacks. They lack long-distance [12] and non-consecutive word interactions and only capture semantic information in local consecutive word sequences. In order to relieve these problems, researchers try to make use of graph neural networks(GNNs) [13]. GNN has a rich relational structure

✉ Peiyu Liu
liupy@sdnu.edu.cn

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

and can preserve global structure information of a graph in graph embedding, and with GNN, long-distance interactions between words could be captured to improve the final text classification performance. GNN also gets wide attention [14]because of its superior performance. There are many models based on GNN, such as Text-Level-GNN, TextGCN, TextING, TensorGCN and so on. However, there are many problems with these graph-based methods. First, the contextual aware word relations are neglected and high memory consumption. For example, TextGCN [15] solves the problem of how to convert text into the graph, but it consumes too much memory and lacks consideration about text-level word interactions [16], leading to the model can not understand the semantic of text well. Second, they cannot exploit rich relational information present among entities in texts. To be specific, TextING [17] simplifies text graph and reduces memory consumption, but it neglects semantic features representation. Similarly, TensorGCN [18] ignores the update of nodes' semantic information, which makes it can not sufficiently mine the semantics, and lacks mastering semantic information.

We propose a new framework, named GText, which can further mine semantic features and relationships within the text. In this work, **first**, we construct a text graph based on semantic features for each document [19] and only contain words as nodes. **Then** we can get a very comprehensive contextual semantic relationship via the SIP(Semantic Information Passing) mechanism, and **finally**, we get text-level representation by a gate mechanism. Our highlights including constructing a semantic features graph for each text to simplify the complexity of graph structure, get semantic relationships, and making use of the SIP mechanism to achieve text information collection, integration, and non-consecutive word interactions, capturing text-level representation by a gate mechanism to improve final text classification performance. In a word, compared with traditional models and above models based on GNN, our model has three contributions as follows:

- Our model builds a semantic features graph for each text, which simplifies the complexity of graph structure, and reduces memory consumption.
- Our approach can achieve long-distance and discontinuous words semantic information interaction and integration. We also establish a deeper relationship representation between words for text-level representation.
- Extensive experiments are conducted on several benchmark datasets to illustrate the effectiveness of GText for text classification.

The remainder of our paper is organized as follows: In Section 2, we describe the related work of text classification. In Section 3, the details of the proposed method are described. In Section 4, we present our experimental results and make the analysis. Finally, we briefly conclude the paper in Section 5.

2 Related work

Natural language processing has always been an important direction in the field of computer science and artificial intelligence, and text classification is a classic problem in natural language processing. In what follows, we briefly review existing studies on text classification methods.

2.1 Traditional text classification methods

People have started the research of text classification in the last century. In the early years of text classification, Naive Bayes [4], KNN [5], SVM [6] are widely used text classification methods. Among them, the Naive Bayesian classifier is a weak classifier. It is easy to build, and suitable for large data sets. However, it has a drawback that the assumption of independent prediction, is almost impossible in real life. And KNN determines the category of the new document according to the similarity of vectors of each document. It is very suitable when classification standards are uncertain, but it needs to compare the new text with all existing training documents when judging the category of text, therefore, the computational cost is very high. Similarly, SVM is also a classical method, it has advantages in solving small samples, however, it also has the problem that computational overhead is relatively large. In a word, these methods usually need at the cost of labor and efficiency, which all affect their performance in the task. With the coming of the information era and the rapid development of the Internet, multimedia information is used widely and deeply, people have higher requirements for text classification, so the deep learning methods are promoted.

2.2 Text classification based on deep learning

For nearly a decade, with the rise of statistical learning methods, a set of deep learning methods are promoted to solve such issues. Among them, neural networks such as RNN and CNN have been widely used in text classification. For example, TextCNN [7] model uses multiple different sizes and kernels to extract key information in sentences, which contributes to better capturing local correlation. However, its convolution and pooling operation will lose the vocabulary order and location information in the text. Similarly, TextRNN [9] model, which uses RNN cyclic neural network to solve the problem of text classification, but it will appear gradient disappearance and gradient explosion, which makes it difficult to learn the long-distance correlation of sequences. In addition, the fastText [11] model is a classical model, by introducing the concept of subword n-gram, the fastText [11] model solves the problems of morphology, low-frequency words, and unregistered words, which makes it can get good results in tasks with a large number of samples and many category labels. However, due to a large number of parameters that need to be estimated, the model may expand and the required memory is too large, which affects the performance of the model. Additionally, there are also many methods to combine neural networks with attention or others, such as ACT [20], MARTA [21], Knowledge-Aware Leap-LSTM [22] , SALNet [23] and so on, there are also methods based on label, for instance, AGN [24], LightXML [25], HTTN [26].

2.3 Text classification with GNN

In recent years, people began to notice the difference of GNN [27]. GNN was originally a neural network that can directly act on images. Recently, there are more and more models based on GNN applying to text classification, such as TextGCN [15] model, which constructs a text map for corpus based on word co-occurrence and word-word semantic relationship, then learns a text GCN for the corpus to improve the accuracy of text classification. However, due to the mapping for the whole corpus, the memory consumption is too large leading to the performance of the model being affected. Similarly, TextING [17] model, which creates a text graph through word co-occurrence, and then classifies the text by summarizing and learning new feature information. However, the text graph created by word co-occurrence can not well represent the semantic relationship between nodes, which will affect the performance of text classification.

Compared with the above models, our model can well relieve these problems. Firstly, our model uses graph structure, which can well solve the long-distance learning problems of TextCNN, TextRNN, and other traditional models. Secondly, our model builds a graph for each text instead of the corpus, and our model only has word nodes, which simplifies the complexity of graph structure and making a model can well solve the problem of excessive memory consumption. In addition, compared with TextING model, our model uses semantic similarity building text graph, which makes the semantics of text expressed better. Finally, we use a gate mechanism for mining the explicit keywords of the whole document, it is an attention mechanism, which can improve the model finding text-level representation power, then improve classification performance.

3 Method

In this section, we will introduce our model in detail. Firstly, we create text graphics based on the semantic features of the text, so that each text has its text-level graphical representation. Then using SIP(Semantic Information Passing) mechanism to ensure that the context semantic features will not be lost. Finally, the attention mechanism selects keywords for the text and classifies text according to the keyword information. The overall framework of our model is shown in Fig. 1.

3.1 Building semantic features graph

In this part, we create the semantic features graph. The semantic features graph algorithm is shown below. Firstly, we extract and pre-process text data, and use embedding to represent word semantic features. After getting the vector representation of the feature words, we start to create the semantic features graph.

Here, we take the only word representation in the text as the vertex of the graph, which is recorded as v.

$$v = \{v|v1, v2, v3...\} \tag{1}$$

If the weight between two-word nodes is greater than a certain value, it is considered that there is an association between them, which means there is an edge. The edge is marked as e, i and j are two-word nodes, ws represents the sliding window size.

$$e = \{e_{ij}|i \subseteq ws, j \subseteq ws\} \tag{2}$$

we use cosine similarity to calculate the semantic similarity between two-word nodes, and the obtained semantic similarity as the weight between word nodes, indicating the degree of dependency between word nodes.

$$similarity = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{3}$$

The similarity explained that the weight is to be calculated in a set sliding window. The sliding window here can be manually adjusted as needed.

Our model builds a semantic features graph for each text instead of the whole corpus, which can not only reduce unnecessary memory consumption but also improve the accuracy of semantic information transmission in the text. Secondly, there are only word nodes in our semantic graph. Using only word nodes can reduce the complexity of the graph, and improve the efficiency of node information dissemination in the graph.

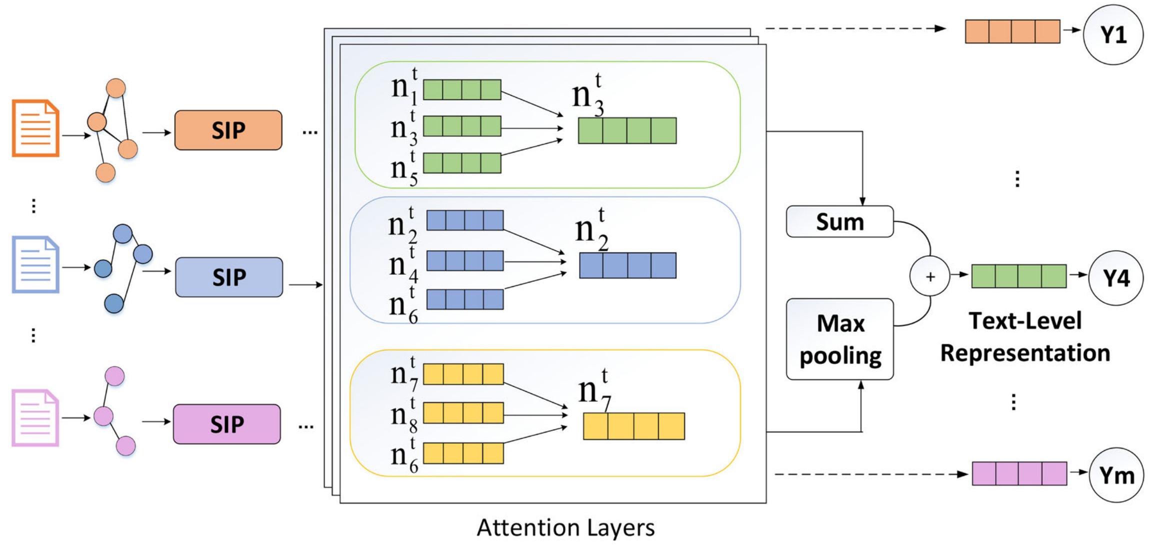


Fig. 1 GText framework. Illustrating the process of text classification by GText. First, we build a semantic features graph for every document, then we feed it into SIP(Semantic Information Passing, it will be described in Section 3.2), and finally we choose text level representation based on attention layers

Algorithm 1 Building semantic features graph.

```
Input: authoritative dataset, texts.
Output: Semantic Features Graph.
Train GText model on the training data of the given task;
Get semantic features with GText model for words in each document of the corpus;
Get the only word representation in the text as the vertex of the graph;
for calculate word-word similarity do
    if word-word similarity reach a certain value then
        word-word similarity as word-word weights,they have edges;
    else
        No semantic relationship between word and word;
return text semantic features graph.
```

3.2 Semantic information passing

After getting the semantic features graph, we set up SIP semantic information transmission mechanism to obtain more comprehensive and accurate semantic information. To ensure that each node in the graph can keep the most valuable semantic information and further transmit it, we ask each node to interact with their neighbor nodes and gain neighbor nodes information, therefore, for each word node in the text graph, they do not exist independently.

$$S = A_{n_i} n_i^t W_s \tag{4}$$

Here, s is the information of all neighbor nodes collected by node n, where A represents the adjacency matrix.

$$\eta = sigmoid(W_\eta + U_\eta + b_\eta) \tag{5}$$

$$a = sigmoid(W_a + U_a + b_a) \tag{6}$$

η and a are important parameters that determine the degree of information retention. They enable nodes to selectively retain the most valuable information, which contributes to updating and optimization of node information in the next step, $\odot$ denotes dot production operation.

$$\lambda = a \odot \eta \tag{7}$$

$$n_i' = tanh(W_{n_i'} + U_{n_i'}\lambda + b_{n_i'}) \tag{8}$$

$$n_i^{t+1} = (1 - \eta) \odot n_i^t + n_i' \odot \eta \tag{9}$$

n_i^{t+1} is a node with accurate semantic information, we obtained it by node n_i^t information update sufficiently. η determines the influence of neighbor nodes on node n_i^t, and it determines the retention degree of node n_i^t on neighbor information. Here, all U, W, b, η are variable parameters. They will be continuously optimized in the training to ensure the effective update of node information, then improving the semantic understanding of word nodes in the text and the subsequent text classification function.

3.3 Classification based on semantic

Through the previous two steps, the nodes in the semantic features graph have been fully updated, therefore, each node has more accurate text semantic information. For the updated nodes, we call them the nodes at t + 1 time, record as n_i^{t+1}. We select the most semantic value node from all these updated nodes by adding an attention mechanism. That means selecting the important text-level representation in the text, then making the final prediction and classification for the text through the selected keyword information at the text. The functional expression is defined as follows:

$$W_n = MLP(n_i^{t+1}) \tag{10}$$

$$h_i = \frac{1}{|v|}\sum_{v\subseteq v} n + Max(n_1^{t+1}...n_i^{t+1}) \tag{11}$$

Where W_n is an attention weight, we use it to represent the significance of words nodes. In addition, we apply a max-pooling function for the text representation, and we average the weighted word features, which makes each word node have an impact on the final result, but the keywords contribute more explicitly.

$$y_i = softmax(h_i W_n + b) \tag{12}$$

$$L = -\sum_i y_{label} log(y_i) \tag{13}$$

Finally, the text-level representation is sent to the softmax layer for final label prediction, and the classification results are obtained.

4 Experiments

In this part, our goal is to evaluate the overall performance of our model GText on two benchmarks datasets under popular evaluation index Test Accuracy. To verify and analyze our model more comprehensively, we will experiment with our model from the aspects of the experimental setting, experimental result analysis, ablation experiment, and parameter sensitivity.

4.1 Datasets

Our experiments are conducted on two public and authoritative standard datasets: film review data set and Ohsumed data set.

MR dataset: MR data set is a classic film review data set. It is a data set used for binary emotion classification. It is useful in multiple text classification models. It mainly divides film reviews into positive reviews and negative reviews, including 5331 negative reviews and 5331 positive reviews. We divided it into tests and training.

0hsumed dataset: Ohsumed data set comes from medline10, a medical information database. It contains the titles or abstracts of 270 medical journals and 348566 documents from 1987 to 1991. We used 13929 unique cardiovascular disease abstracts out of 20000 before 1991, with case categories from 23 disease types in each document. When performing single-label classification, multi-label documents belonging to multiple classes will be excluded, leaving only 7400 documents belonging to one category, including 3357 documents in the training set and 4043 documents in the test set.

We will show specific information about our dataset on Table 1

4.2 Baselines

In order to make a comprehensive evaluation for our model, we compare our text classification model GText with several recognized text classification models with good performance.

RNN [9]: RNN uses the last hidden state as the representation of text. RNN recurrent neural network is used to solve the problem of text classification and try to infer the label or label set of a given text (sentence, document, etc.). Such as emotion analysis, news topic classification, false news detection, etc.

CNN [7]: CNN employs convolution and maximum pool operations are performed on word embedding to obtain the representation of text.

fastText [11]: fastText, average word or n-gram embedding is used as document embedding. fastText combines the most successful concepts in natural language processing and machine learning. These include the use of word bag and n-gram bag to represent statements, the use of subword information, and the sharing of information among categories through hidden representation.

SWEM [28]: It is a simple word embedding model, which employs simple pooling strategies operated over word embeddings.

TextGCN [15]: TextGCN model converts text to graph and using GCN learn to text and text classification.

TensorGCN [18]: TensorGCN model establishes a text map according to the three aspects of text semantics, word order, and grammar integrates and summarizes the three maps, strives to accurately understand the semantics of the text, and improves the efficiency of text classification.

TextING [17]: In the TextING model, text classification is based on text graphs, so that text classification has the ability of inductive learning and improves the efficiency of text classification.

Table 1 Summary statistics of dataset

Dataset	#Docs	#Train	#Test	#Classes	#Words	#Average Length
MR	10,662	7,108	3,554	2	18,764	20
Ohsumed	7,400	3,357	4,043	23	14,157	136

4.3 Settings

In this section, we will introduce some details of our experiment. We use the glove word embedding method, and the data dimensions we enter are all 300 dimensions. For all data sets, we give the training set and test set and divide the training set into the actual training set and verification set according to the ratio of 9:1. Moreover, we set the learning rate to 0.001 and dropout to 0.5. For baseline models, we all use default parameter values, just as they were in the original paper or implementation. Our model runs under the TensorFlow framework, and the test accuracy is used as the evaluation index. Compared with other methods, our model has achieved the latest results under this evaluation index.

4.4 Experimental results and analysis

In this part, we will show our specific experimental results and analyze the results. As shown in Table 2, compared with other models, our model is almost always superior to other baseline models in MR and Ohsumed datasets. And in most cases, our model performs better than the strongest baseline model. From Fig. 2, we can more intuitively see the differences between them.

We find that compared with our model, the experimental results of traditional neural network models such as RNN and CNN are generally poor. This is because these models give priority to the order and local information of the text and ignore the global semantic information of the text. In addition, these neural network models can not carry out long-distance information transmission. In our model GText, a node can have multiple neighbor nodes, and the neighbor nodes are no longer limited to several nearby nodes, they may not have to be adjacent. Moreover, through the information transmission between neighbor nodes, the semantic features information can be well transmitted in the global scope, and the text-level semantic information can be expressed better expression. However, neural network models such as CNN and RNN do not have such structural functions. In addition, the experimental results in the figure above further show that our model is superior to other baseline models based on GNN in terms of test accuracy.

Table 2 Text accuracy comparison with baselines on benchmark datasets

Model	MR	Ohsumed
CNN-rand	0.7498±0.0070	0.4387±0.0100
CNN-non-static	0.7775±0.0072	0.5844±0.0106
RNN	0.7768±0.0086	0.4927±0.0700
fastText	0.7514±0.0200	0.5770±0.0490
SWEM	0.7665±0.0063	0.6312±0.0055
TextGCN	0.7674±0.0020	0.6836±0.0056
TensorGCN	0.7791±0.0007	0.7011±0.0024
TextING	0.7832±0.0020	0.7042±0.0039
Our Model	0.8007±0.0028	0.7940±0.0014

In the TextGCN model, the text is successfully converted into a graph, but it is based on the whole corpus, which leads to a large amount of storage space consumption and can not support online testing. Our model is to build a text graph for every text. Building a graph for each document can not only avoid excessive resource consumption but also increase the semantic understanding of the text.

In the TensorGCN model, semantics, word order, and grammar are taken into account in text mapping, but too many factors in the process of mapping will increase the complexity of the model and the time delay, and reduce the work efficiency of the model. The ultimate purpose of text classification is to put the text into the corresponding label, so the model should focus on text semantics. Our model mainly focuses on the semantics of text, paying attention to the understanding and transmission of text semantic information.

In the TextING model, using the word co-occurrence method to build the graph for the text. Word co-occurrence is to calculate the frequency of a group of words in the text as their similarity, but the method of word co-occurrence in text classification is not so applicable. Because sometimes not all phrases with high frequency can represent the semantics of the text itself. Our model uses cosine similarity to calculate all the keywords that may represent the theme of the text, and then uses the attention mechanism to select the most representative as the basis of classification, which can contribute to selecting the text-level semantic representation and improve the efficiency of text classification.

From the above figure, we also can see that our model performance better than other baseline models based on GNN. This is because our model can better understand and transmit the text semantics than other baseline models based

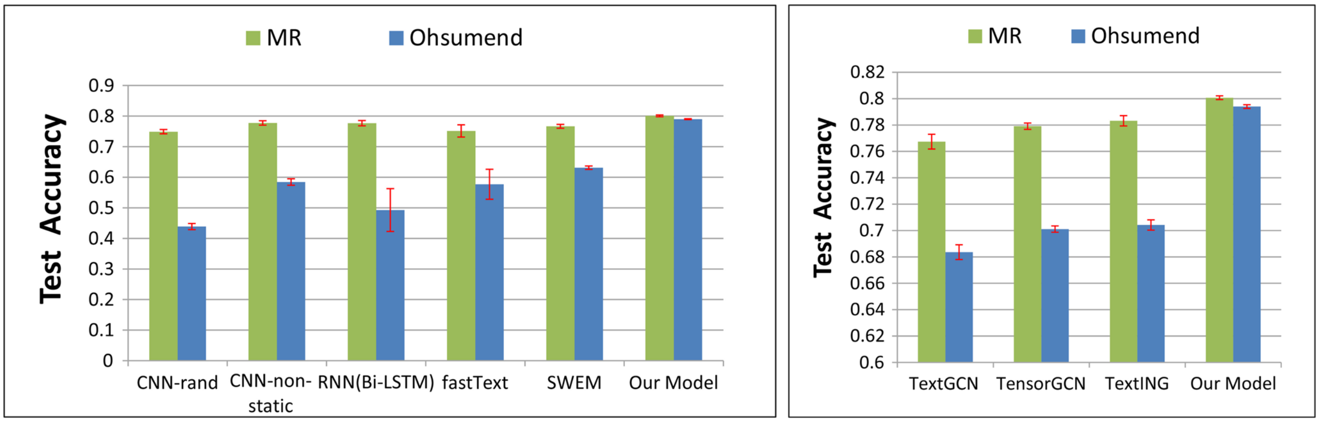


Fig. 2 The left of figure is test accuracy with different tradtional models,and the right of figure is test accuracy with different models based on GNN. The red brackets on each columnar door are the range of value changes

on GNN, and our model has better semantic expression ability. In addition, we notice that our model performance is better in the Ohsumed data sets. This may be because MR data sets are short text and the density of text graphs is low, therefore, our text semantic features graph and SIP information transmission mechanism have little impact on it. But for the Ohsumed data sets, complex long sentences, a large number of different words all can give full play to the understanding and transmission function of our model. Finally, improving the efficiency of model text classification.

4.5 Ablation experiment

In order to further study the influence of each part of GText on its model performance, we also designed several ablation experiments.

It can be seen from Fig. 3 that the performance of the GText ablation experiment is obviously inferior to GText. The experimental results show that these discarded modules have a significant impact on the performance of the model.

After removing the semantic information module, the model becomes w/o semantic. We can observe the performance changes of the model on the two data sets from Fig. 3. The performance of w/o semantic on both MR data sets and Ohsumed data sets has declined, which shows that semantic mapping has an important impact on the performance of the model. A good semantic map can connect truly semantically related words together, and correctly define their correlation degree so that the model can fully understand their relationship, reduce the burden for subsequent modules and increase the accuracy of text classification.

The w/o SIP is a model formed after removing the SIP information transmission mechanism in the GText model. From the above figure, it can be observed that the test accuracy of the model without the SIP module is greatly reduced on the MR and Ohsumed, which fully illustrates the importance of the SIP mechanism. SIP mechanism can flexibly spread messages in text, and the message can be learned and retained effectively according to the semantic learning and understanding of the text, which not only reduces the unnecessary waste of resources but also increases the work efficiency of the model and improves the accuracy of text classification.

After removing the attention mechanism module in the GText model, the w/o attention model is formed. It can be observed from Fig. 3 that the performance of the model will decline to some extent after removing the attention module. The attention mechanism can help the model better select the keywords semantic features in text and improve the test accuracy of the model in text classification.

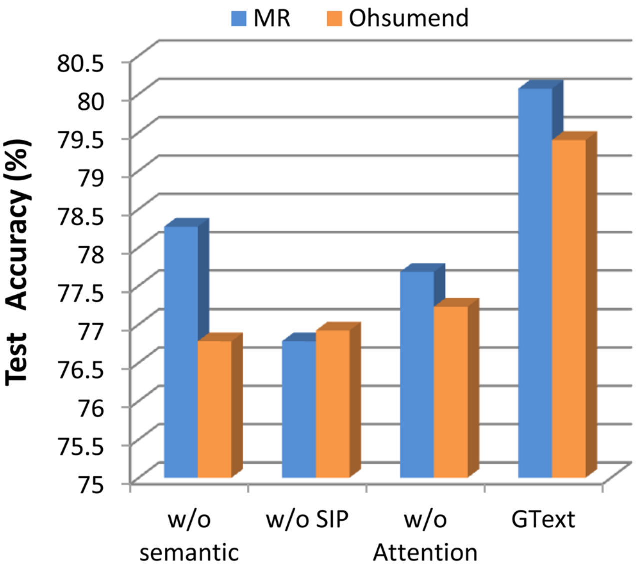


Fig. 3 The test accuracy of GText in Ablation Experiment

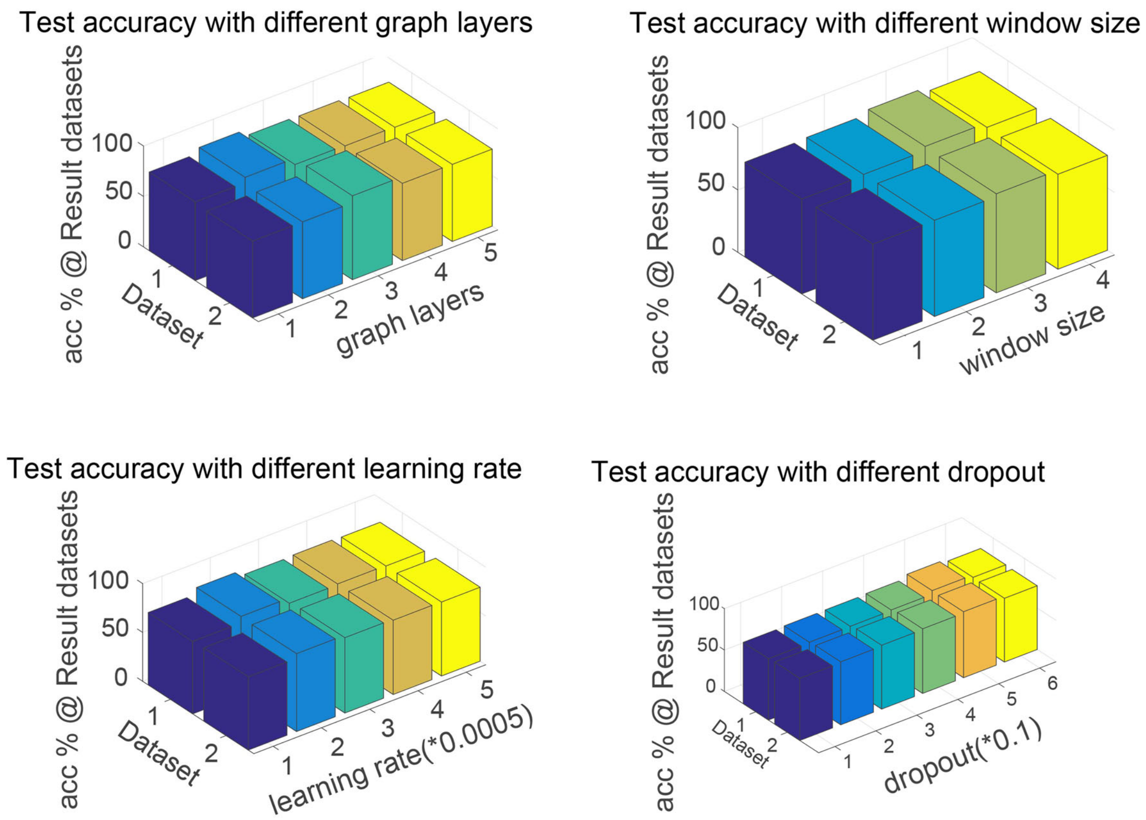


Fig. 4 Impact of the parameter graph layers, window size, learning rate and dropout on MR and Ohsumed datasets. The 1 and 2 of the dataset-axis represent MR and Ohsumed, respectively. The main contrast is the influence of these parameters on test accuracy

4.6 Parameter sensitivity

The model performance on MR and Ohsumed with different parameters is reported in Fig. 4. Notably, for parameter graph layers, the best performance of our model is achieved when the graph layer is 3 for MR and the graph layer is 4 for Ohsumed. It indicates that with the increment of the graph layers, nodes can get more neighbor information. However, when the number of graph layers reaches a certain value, the test accuracy starts to decline. That means the learning ability of nodes is limited.

Figure 4 exhibits the performance of GText with different window sizes. With the increment of window size, the test accuracy of our model on MR and Ohsumed also increased. When the test accuracy reaches a peak value, it begins to decline. It illustrates the performance is also affected by the window size.

Figure 4 also exhibits the performance of GText with a varying learning rate on MR and Ohsumed. The result reveals that with the increment of the learning rate, the GText model learns more and more semantic information. Nevertheless, the situation reverses with a continuous increment, where the model reaches the fitting point in advance. In addition, Fig. 4 also illustrates the performance as well as the text accuracy of GText with a varying dropout on MR and Ohsumed. It presents a similar trend as the learning rate when the value of dropout increases.

5 Conclusion

For text classification, previous researches focus on the locality of words and ignore text-level word interactions. In this paper, inspired by how a human being understands a text and acquires knowledge, we fully mine semantic features and relationships from multiple perspectives. And experiments results show that our model is superior to the best baseline. We build a semantic feature graph for each document separately to get the semantic relationship between word nodes. Each node can exchange information with neighbor nodes to achieve the transmission of semantic information, which can better contact the context information. Our model highlight as follows: First, we achieve fine-grained text-level word interaction. Second, we obtain more comprehensive semantic information. Third, experiments show that our model has certain advantages in context semantic transmission and semantic information selection. We set some parameter values and use these parameters to determine the impact of a node's neighbor node information, and then improve the accuracy of semantic information in the model, ensuring that the key semantic information is retained, and make the most effective semantic information play a role in text classification. Then improving the accuracy of text classification and the efficiency of the model.

Acknowledgements This work was supported in part by the National Social Science Foundation under Award 19BYY076, in part Key R & D project of Shandong Province 2019JZZY010129, and in part by the Shandong Provincial Social Science Planning Project under Award 19BJCJ51, Award 18CXWJ01, and Award 18BJYJ04.

References

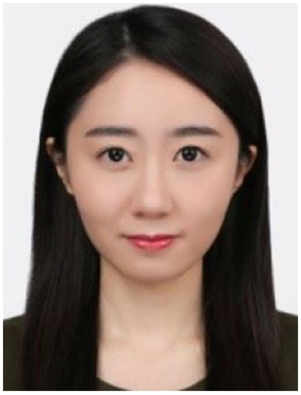
1. Wang Y, Huang M, Zhu X, Zhao L (2016) Attention-based LSTM for aspect-level sentiment classification. In: Su J, Carreras X, Duh K (eds) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. The Association for Computational Linguistics, pp 606–615
2. Wang SI, Manning CD (2012) Baselines and bigrams: Simple, good sentiment and topic classification. In: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers. The Association for Computer Linguistics, pp 90–94
3. Lipton ZC, Kale DC, Elkan C, Wetzel RC (2016) Learning to diagnose with LSTM recurrent neural networks. In: Bengio Y, LeCun Y (eds) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings
4. Androutsopoulos I, Koutsias J, Chandrinos K, Paliouras G, Spyropoulos CD (2000) An evaluation of naive bayesian anti-spam filtering. arXiv:0006013
5. Tan S (2006) An effective refinement strategy for knn text classifier. Expert Syst Appl 30(2):290–298
6. Forman G (2008) Bns feature scaling: An improved representation over tf-idf for svm text classification. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08. Association for Computing Machinery, New York, pp 263–270
7. Kim Y (2014) Convolutional neural networks for sentence classification. In: Moschitti A, Pang B, Daelemans W (eds) Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, october 25-29, 2014, doha, qatar, A meeting of sigdat, a special interest group of the ACL. ACL, pp 1746–1751
8. Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S (2010) Recurrent neural network based language model. In: Kobayashi T, Hirose K, Nakamura S (eds) INTERSPEECH 2010, 11th annual conference of the international speech communication association, makuhari, chiba, japan, september 26-30, 2010. ISCA, pp 1045–1048
9. Liu P, Qiu X, Huang X (2016) Recurrent neural network for text classification with multi-task learning. In: Kambhampati S (ed) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. IJCAI/AAAI Press, pp 2873–2879
10. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Bonet B, Koenig S (eds) Proceedings of the twenty-ninth AAAI conference on artificial intelligence. AAAI Press, Austin, pp 2267–2273
11. Joulin A, Grave E, Bojanowski P, Mikolov T (2017) Bag of tricks for efficient text classification. In: Lapata M, Blunsom P, Koller A (eds) Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Volume 2: Short Papers. Association for Computational Linguistics, Valencia, pp 427–431
12. Peng H, Li J, He Y, Liu Y, Bao M, Wang L, Song Y, Yang Q (2018) Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In: Champin P-A, Gandon F, Lalmas M, Ipeirotis PG (eds) Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, address=Lyon. ACM, pp 1063–1072
13. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net
14. Cai H, Zheng VW, Chang KC-C (2018) A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Trans Knowl Data Eng 30(9):1616–1637
15. Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019. AAAI Press, Honolulu, pp 7370–7377
16. Li Z, Cui Z, Wu S, Zhang X, Wang L (2019) Fi-gnn: Modeling feature interactions via graph neural networks for CTR prediction. In: Zhu W, Tao D, Cheng X, Cui P, Rundensteiner EA, Carmel D, He Q, Yu JX (eds) Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, address=Beijing. ACM, pp 539–548
17. Zhang Y, Yu X, Cui Z, Wu S, Wen Z, Wang L (2020) Every document owns its structure: Inductive text classification via graph neural networks. In: Jurafsky D, Chai J, Schluter N, Tetreault JR (eds) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020. Association for Computational Linguistics, pp 334–339
18. Liu X, You X, Zhang X, Wu J, Lv P (2020) Tensor graph convolutional networks for text classification. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020. AAAI Press, New York, pp 8409–8416
19. Rousseau F, Kiagias E, Vazirgiannis M (2015) Text categorization as a graph classification problem. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, Volume 1: Long Papers. The Association for Computer Linguistics, Beijing, pp 1702–1712
20. Li P, Zhong P, Mao K, Wang D, Yang X, Liu Y, Yin J, See S (2021) ACT: an attentive convolutional transformer for efficient text classification. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event. AAAI Press, pp 13261–13269
21. Arous I, Dolamic L, Yang J, Bhardwaj A, Cuccu G, Cudré-Mauroux P (2021) MARTA: leveraging human rationales for explainable text classification. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event. AAAI Press, pp 5868–5876
22. Du J, Huang Y, Moilanen K (2021) Knowledge-aware leap-lstm: Integrating prior knowledge into leap-lstm towards faster long

text classification. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event. AAAI Press, pp 12768–12775

23. Lee JH, Ko S-K, Han Y-S (2021) Salnet: Semi-supervised few-shot text classification with attention-based lexicon construction. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event. AAAI Press, pp 13189–13197
24. Li X, Li Z, Xie H, Li Q (2021) Merging statistical feature via adaptive gate for improved text classification. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event. AAAI Press, pp 13288–13296
25. Jiang T, Wang D, Sun L, Yang H, Zhao Z, Zhuang F (2021) Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event. AAAI Press, pp 7987–7994
26. Xiao L, Zhang X, Jing L, Huang C, Song M (2021) Does head label help for long-tailed multi-label text classification. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event. AAAI Press, pp 14103–14111
27. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi VF, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R, Gülçehre C, Song HF, Ballard AJ, Gilmer J, Dahl GE, Vaswani A, Allen KR, Nash C, Langston V, Dyer C, Heess N, Wierstra D, Kohli P, Botvinick M, Vinyals O, Li Y, Pascanu R (2018) Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261
28. Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L (2018) Joint embedding of words and labels for text classification. In: Gurevych I, Miyao Y (eds) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers. Association for Computational Linguistics, Melbourne, pp 2321–2331

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yahui Li was born in 1994. She is pursuing the master's degree in the College of Computer Science, Shandong Normal University, China. Her research focuses on text classification.



Yifan Liu received the Master degree from The Chinese University of Hong Kong in 2012. She is currently an associate professor, master supervisor, with the School of International Education, Shandong Normal University, China. Her research interests include Applied Linguistics, Computational Linguistics and Natural Language Processing.

Zhenfang Zhu received the Ph.D. degree from Shandong Normal University in 2012, and he was a postdoctoral fellow at Shandong University between 2012 and 2015. He is currently an Associate Professor and a Master's Supervisor with the School of information science and electrical engineering, Shandong Jiao Tong University, China. His research interests include network information security, natural language processing, and applied linguistics.

Peiyu Liu received the master's degree from East China normal university in 1986.He is currently a Second-level professor, doctoral supervisor, with the School of information science and engineering, Shandong Normal University, China. He is the national outstanding science and technology worker, middle-aged and young expert with outstanding contribution in Shandong province, famous teacher of Shandong province. His research interests include network information security, information retrieval, natural language processing, and artificial intelligence.