A BERT-based Heterogeneous Graph Convolution Approach for Mining Organization-Related Topics

Haoda Qian[1,2] Minjie Yuan[1,2] Qiudan Li*[1] Daniel Zeng[1]
[1]The State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
{qianhaoda2019, yuanminjie2021, qiudan.li, dajun.zeng}@ia.ac.cn

***Abstract*—Mining organization-related topics is helpful to analyze the information dissemination situation. Existing methods based on graph neural networks mainly consider the association between words and documents, they ignore the semantic interactions between documents, and do not consider the heterogeneity of edges which are difficult to solve the challenge of blurred topic boundaries in real scenarios, resulting in performance loss. This paper proposes a BERT-based Heterogeneous Graph Convolution Network (BERT-HGCN) approach for semi-supervised topic mining that comprehensively considers multi-semantic relations between words and documents. It deeply combines the advantages of transductive learning with pre-training models. We model documents as graph-structured data and capture multiple semantic dependencies among word-word, word-doc, and doc-doc via information propagation mechanism. During the model learning process, a two-stream encoding mechanism is used to learn the structural and semantic representations, which combines a hierarchical graph convolution network (HGCN) and a BERT-based auto-encoder. It considers both edges heterogeneity and semantics of original documents. Finally, a dual-supervision loss is used to train the classifier based on graph nodes and semantic representations for topic mining. We empirically evaluate the performance of the proposed model on a real-world organization-related dataset, and the experimental results demonstrate the efficacy of the model.**

Keywords—topic mining, multi-level semantic, heterogeneous graph neural network, pre-training model

I. INTRODUCTION

With the rapid development of network technology, social platforms and web media have become important channels for organizations to convey information. According to a McKinsey Quarterly report, 50% of the more than 1,700 organizations surveyed use social networking, 41% use blogs, 25% use wikis, and 23% use microblogs [1]. As of 2021, over 90% of organizations are using social platforms and web media to release news according to The Global Media Intelligence Report 2021[1]. Mining organization-related topics can help analyze the situation and influence of information dissemination. In practice, the content of organizational reports contains multi-perspective expressions at the word and document levels, usually covering multiple topics with different emphases. There exists an overlap between topics, which makes it difficult to distinguish the shallow semantics of a single element. For topic analysis, it is not enough to make accurate classifications only from the connections of word and document. Mining semantic associations at a deeper level is necessary. Therefore, jointly modeling the connection of word-doc and original semantic information to analyze organization-related topics can get deeper insights into the mechanism of information dissemination, which mines differences of topics and leads to a better-informed decision.

Existing supervised topic mining methods usually follow a two-step paradigm. They first capture the semantics of each document by sequence separately. Next, they classify documents into different topics via learned text representations. [2] proposed a word-level supervised model, which introduced a class-dependent linear transformation on the topic mixture proportions. [3] employed LSTM to capture long-range semantic information. [4] introduced an attention mechanism to enhance semantic interaction. However, the above work cannot capture interactions between documents and words simultaneously, which leads to unsatisfactory performance.

Modeling texts as graph can bring interactive information from documents and words. Existing graph neural network based methods mainly focus on mining the structure relationship between words and documents. [5, 6] considered words and documents as nodes, and semantic relationship between word and doc as edges. The node representations are learned by graph convolutional networks. Considering the word-level and doc-level semantic connections, [7] constructed a single heterogeneous graph neural network on multilingual corpus. In addition, [8] used a pre-trained language model to improve the representation ability and facilitate downstream tasks. However, most of the existing methods ignore the edge heterogeneity and order-of-magnitude differences of edge weight, and pay little attention to the semantics of the documents themselves. The challenge lies in how to effectively incorporate the relationship between texts and pre-trained semantic representations to discover the distribution of topics from multiple perspectives.

This paper proposes a heterogeneous graph neural network based topic mining model, which deeply fuses the advantages of graph-based transductive learning and pre-training models. The constructed graph takes words and documents as nodes, multiple semantic interactions among word-word, word-doc, and doc-doc are regarded as edges. Specifically, according to the three types of semantic associations, the weights of the edges are calculated in seperate graph networks. During the model

* Corresponding Author

[1] https://www.emarketer.com/content/global-media-intelligence-report-2021

learning process, a two-stream encoding mechanism is used to learn the structural and semantic representations, which combines a hierarchical graph convolution network (HGCN) [9] and a BERT-based [10] DNN auto-encoder. It simultaneously considers edges heterogeneity and semantics of original documents. Then, the model performs linear interpolation between the two streams to obtain a more comprehensive representation. Finally, topic mining is performed on graph node representations and semantic representations, respectively, and a dual-supervised loss function is used to jointly optimize the model.

We empirically evaluate the performance of the proposed model on a real-world organization-related dataset, the experimental results demonstrate that combining the deep relationship between texts and pre-trained semantic representations contributes to mining topics accurately.

In summary, our main contributions are as follows:

- We model documents and words as graph-structured data and design a heterogeneous based framework to explore the complex semantic relations.
- The proposed model deeply fuses the relationship between documents and semantic representation to mine topics from multiple perspectives, which considers the node interactions of the graph and the contextual semantic information simultaneously.
- We demonstrate the efficacy of the model on a real-world dataset, the quantitative and qualitative results contribute to a more accurate analysis of organization information dissemination.

The rest of this paper is organized as follows: In section II, we review related works. The detailed components of our proposed model are presented in section III. Then, the experiments on a real-world dataset are discussed in section IV. Finally, we summarize our work and put forward future research directions in section V.

II. LITERATURE REVIEW

Our work is related to topic mining, text representation learning and graph-based text classification. In this section, we review the related works.

A. Topic Mining

Existing topic mining methods can be divided into unsupervised and supervised methods. Unsupervised topic models, such as pLSA [11] and LDA [12], exploited modeling relationships between texts to discover latent topics. Supervised methods used sequential models to encode semantic information, and then adopted inductive methods to classify texts into different topics. DiscLDA[2]introduced a class-dependent linear transformation on the topic mixture proportions. CatE[13] developed a category-name guided text embedding method for discriminative topic mining. [14] jointed spherical tree and text embedding to guide the hierarchical topic discovery process. In summary, the above methods either worked in an unsupervised manner based on text relationships or employed sequential models to embed semantics for topic mining.

Different from the above works, we model the documents and words as nodes and construct heterogeneous graph, which captures multiple semantic dependencies between documents and words via constructed graph structures, and obtain text representations through graph structure learning.

B. Text Representation Learning

Text representation learning has been extensively studied. Effective text representation helps distinguish topics more comprehensively and accurately.

The traditional vector space model [15] used TF-IDF [16] to measure the importance of words, which does not consider the semantics between words. Considering that word embeddings are not specifically optimized for representing sentences, [17] proposed Siamese CBOW to learn word embeddings directly. Many approaches used convolutional neural network (CNN) [18] and recurrent neural network such as long short-term memory network (LSTM) [3] to learn text representation. Since CNN and recurrent neural network s are unable to distinguish the influence of each part of input sequences, [5] introduced an attention mechanism to solve this problem. Considering the document structure, [5] proposed a hierarchical attention network for document classification, in which both word-level and sentence-level attention were used to capture important words and sentences through learning procedure. With the improvement of computation and storage, large-scale pre-trained language models [10, 19] were proposed for text representation. These models learn syntax and semantic knowledge from large web corpora and achieve a boost in a variety of downstream tasks.

C. Graph-based Text Classification

Semi-supervised topic mining tasks are closely related to transductive text classification. [20] converted text classification into node classification to mine relationships between nodes and learn node representations through label propagation. [18] used documents and words as nodes to construct a heterogeneous graph, and adopted graph convolutional network to learn node representations, which were taken as initialization for downstream tasks. To alleviate the problem of semantic sparseness in short texts, [6] utilized topic and entity information for knowledge supplementation, and proposed a dual-layer graph attention network that can learn the influence of different nodes and edge type. [7] constructed a heterogeneous graph neural network from multilingual corpus by considering the relationship among doc-word and doc-doc, and then applied heterogeneous graph convolution to fuse these information. On the basis of graph representation learning, [8] initialized the node representation with the help of a pre-trained language model, which greatly improves the performance of text classifications .

Most of the existing topic analysis approaches focused on mining doc-word semantic relationships or embedding their own representations of documents. Based on the above research works, we propose a heterogeneous graph neural network based topic mining model, which incorporates these two kinds of information to obtain deep semantic representation for topic mining.

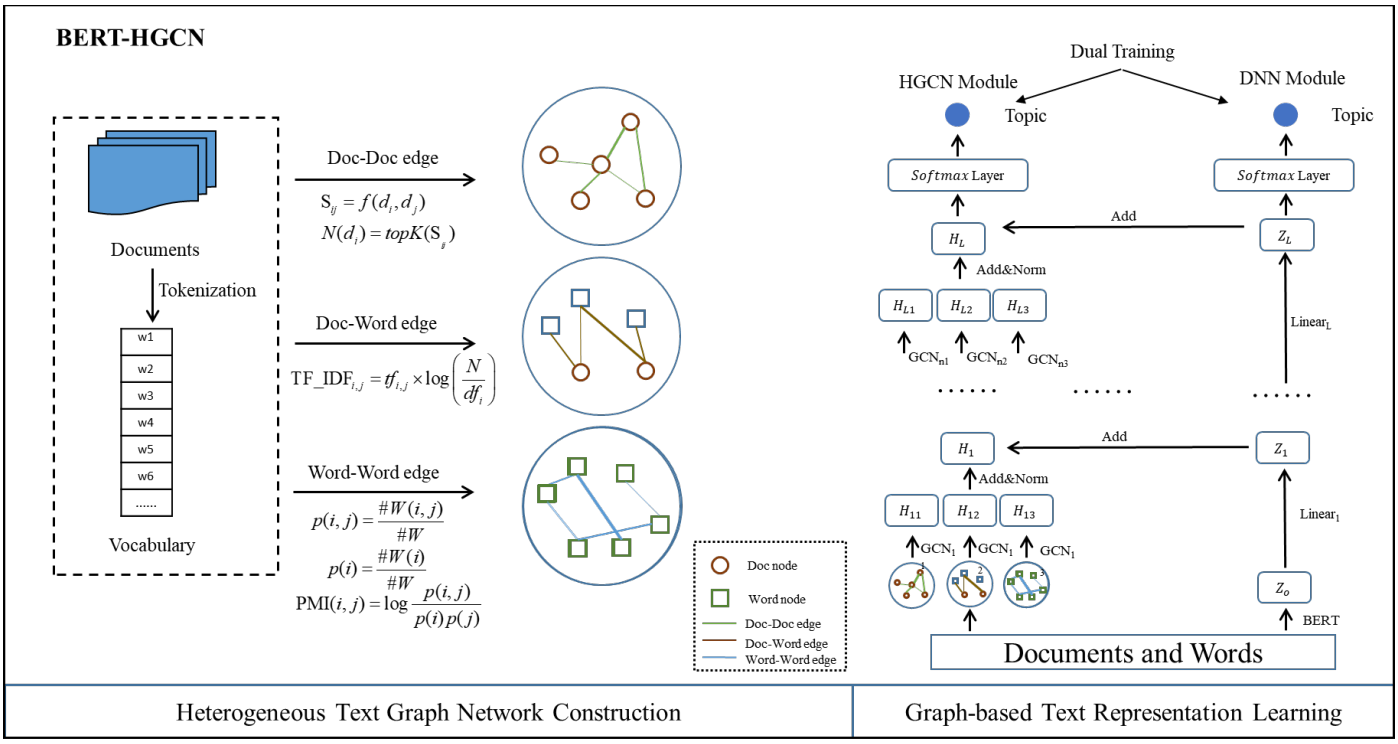

Fig. 1. Framework of our proposed model for topic mining

III. Model for Topic Mining

A. Problem Definition and Formalizations

Semi-supervised topic mining can be modeled as a text classification task. Given m documents, the model will mine the topic $y_i \in Y$ for each document d_i, where Y is a predefined set of event-based topic categories (such as academic conferences, science lectures, etc.). Based on the graph neural network, the document and the words contained in the document are constructed into a text graph network $G = (V, E)$, which formulates the topic mining problem into a node classification paradigm. The text graph contains a node set V, which is composed of the document set $D = \{d_1, d_2, \ldots, d_m\}$ and the word set $W = \{w_1, w_2, \ldots, w_n\}$, where m is the number of document and n is the number of word. The edge sets E describe the semantic relation between nodes V ,including document-document edges $E_{doc-doc} = \{e_{d_1,d_1}, \ldots, e_{d_1,d_m}, \ldots, e_{d_m,d_m}\}$, document-word edges $E_{doc-word} = \{e_{d_1,w_1}, \ldots, e_{d_1,w_n}, \ldots, e_{d_m,w_n}\}$ and word-word edges $E_{word-word} = \{e_{w_1,w_1}, \ldots, e_{w_1,w_n}, \ldots, e_{w_m,w_n}\}$.

B. System Architecture of the Proposed Model

The overview of our model is shown in Fig. 1. Our proposed model provides a unified framework, which is divided into two steps: heterogeneous text graph network construction and graph-based text representation learning.

1) Heterogeneous text graph network construction

In order to capture multiple semantic relations between texts from different perspectives, we first transform the original documents into heterogeneous graphs. As shown in the left part of Fig. 1, the heterogeneous graph network contains two kinds of nodes and three different kinds of edges.

a) Nodes representation

The nodes of the graph include words and documents . Word-level nodes $W = \{w_1, w_2, \ldots, w_m\}$ are represented by words in the vocabulary, which are collected from the documents, while doc-level nodes $D = \{d_1, d_2, \ldots, d_m\}$ denote the documents.

b) Doc-Word interactions calculation

To capture the relation between documents and words, document-word co-occurrence statistics are used to describe the connection between documents and words. According to [17], term frequency-inverse document frequency (TF-IDF) is an effective indicator to measure the importance of words in documents, so we take the TF-IDF value as the weight of the document-word edge.

c) Word-Word interactions calculation

The co-occurrence frequency between words is a commonly used indicator to describe the semantics between words. In topic mining tasks, counting co-occurrence words is a good statistical method. To measure the co-occurrence information between words, a fixed-size sliding window is used to collect the co-occurrence frequency of words across all documents. The weights of word-word edges are represented by point-wise mutual information (PMI), a statistical indicator describing the relationship between words. The PMI can be obtained by formulas (1)-(3), where $\#W(i)$ denotes the number of sliding windows in the corpus containing word i, $\#W(i,j)$ denotes the number of sliding

windows that contain both word i and word j, and $\#W$ denotes the number of all sliding windows in the corpus.

$$PMI(i,j) = log\frac{p(i,j)}{p(i)p(j)} \quad (1)$$

$$p(i,j) = \frac{\#W(i,j)}{\#W} \quad (2)$$

$$p(i) = \frac{\#W(i)}{\#W} \quad (3)$$

A positive PMI value indicates a strong semantic connection between words in the corpus. Therefore, only word-word edges with a positive PMI value are constructed.

d) Doc-Doc relationship mining

The relationship between documents and words alone is not enough to determine the topic to which a document belongs, so the semantic similarity between documents should also be considered. Given an embedding representation of two documents, their similarity can be measured by cosine similarity. During edge construction, we employ local connections to ensure that similar documents can interact. In particular, we find the documents with the top-K cosine similarity from d_i, and connect these documents to d_i by weights.

To sum up, the edge weight between nodes can be expressed by formula (4).

$$A_{ij} = \begin{cases} COS(i,j), & i,j\ are\ documents \\ PMI(i,j), & i,j\ are\ words, PMI(i,j) > 0 \\ TF-IDF(i,j), & i\ is\ document, j\ is\ word \\ 1, & i = j \\ 0, & otherwise \end{cases} \quad (4)$$

Taking the established graph structure and node embeddings as input, we design a heterogeneous graph-based model to better capture complex semantics and obtain highly relevant semantic structures.

2) Node representation learning

Learning a good text representation is crucial for topic mining. To fully utilize contextual information, we use a BERT-based deep neural network (DNN) autoencoder to learn document representations. To model the semantic dependency between documents and words, the text representation learning process is treated as graph node learning. Then, the model adds the output of DNN to the corresponding HGCN layer to merge the two representations. In this way, we are able to leverage the complementary strengths of pre-training models and graph models.

a) DNN module

The DNN module uses a BERT-based autoencoder to learn text representations for documents and words. Assuming that there are L layers of self-encoders, l represents one of the layers of self-encoders. Each layer's output of encoder can be obtained by formula (5). Where $\sigma(\cdot)$ denotes the nonlinear activation function, $W_e^{(\ell)}$ and $b_e^{(\ell)}$ are the weight matrix and bias term of the first layer. The input $Z^{(0)}$ of the first layer is the BERT-encoded or RoBERTa-encoded documents and words representation of X.

$$Z^{(l)} = \sigma\left(W_e^{(l)} Z^{(l-1)} + b_e^{(l)}\right) \quad (5)$$

b) GCN module

Node representation learning for heterogeneous text graph networks aims to gather document and word information through different types of edges. The module first maps document or word into a high-dimensional vector h_i by pre-trained language model, then applies heterogeneous graph neural network to merge useful information from their neighbors. Since different types of nodes contain different types of edges (e.g. document nodes contain word-document edges and document-document edges while word nodes contain word-document edges and word-word edges), fusing information directly in a graph convolutional neural network suffers from decreased performance. Heterogeneous graph convolution network tackles the above shortcomings by fusing the information from various edges through multi-channel graph convolutional neural networks.

Given an undirected graph $G = (V, E)$, let A be its adjacency matrix. In order to avoid the degree of the node being zero, the adjacency matrix becomes $A' = A + I$ after the self-edge is introduced, and the degree matrix is M, and $M_{ii} = \sum_j A'_{ij}$. Then, formula (6) shows the information propagation of each GCN layer.

$$H^{(l+1)} = \sigma\left(\tilde{A} \cdot H^{(l)} \cdot W^{(l)}\right) \quad (6)$$

where $\tilde{A}$ represents the symmetric normalized adjacency matrix, $W^{(l)}$ is the hierarchical linear transformation matrix, $\sigma(\cdot)$ represents the activation function, $H^{(l+1)}$ represents the hidden state of the output of each layer. In particular, the input of the first layer is $H^{(0)} = X$.

The heterogeneous graph convolutional neural network uses different types of edges to divide the graph into multiple subgraphs. It firstly performs a convolution operation within the subgraph, then aggregates the convolution results. The whole process can be expressed by formula (7).

$$H^{(l+1)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \tilde{A}_\tau \cdot H_\tau^{(l)} \cdot W_\tau^{(l)}\right) \quad (7)$$

$\tilde{A}_\tau$ represents a sub-matrix of the normalized adjacency matrix, its edges represent all nodes in the graph, and its columns represent all neighbor nodes connected by edges of type τ. The node representation at layer l is obtained by aggregating the representations of neighbor nodes' different types. Consider different feature spaces and map them to a common feature space. In particular, each type of representation is initialized by the pretrained language model's text representation .

c) Interlayer fusion

Considering that the features learned by DNN are completely derived from the features of documents and words themselves, while the representations learned by GCN

modules come from graph networks, we fuse the representations of DNN modules and GCN modules to generate a more comprehensive and powerful representation. We can see the fusion as formula (8), where ϵ is a balance coefficient, which is set to 0.7 in the experiment, indicating that the merged representation is more dependent on the representation learned by GCN.

$$\widetilde{H}^{(l-1)} = \epsilon H^{(l-1)} + (1-\epsilon) Z^{(l-1)} \tag{8}$$

As shown in formula (9), $\widetilde{H}^{(\ell-1)}$ is taken as the input of the l-th layer of HGCN to generate the representation $H^{(\ell)}$.

$$H^{(l)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \tilde{A}_\tau \cdot \widetilde{H}_\tau^{(l-1)} \cdot W_\tau^{(l-1)}\right) \tag{9}$$

From above, it can be seen that the feature representation from the DNN module is transferred through the normalized adjacency matrix. Since the each layer's representation learned of DNN is different, it is necessary to fuse the each layer's feature representation of DNN into the corresponding HGCN layer.

3) Topic mining

Based on the node embedding learning process, we obtain representations of documents. To mine topics, we connect a classification layer after DNN and GCN respectively. Finally, a dual-supervised loss function is used to jointly optimize the model parameters.

The last layer of the DNN module is a classification layer with a nested softmax function. The loss function for topic classification can be expressed as formulas (10) and (11). Where C denotes the number of topic categories and $\mathrm{D}_{\mathrm{train}}$ denotes the subscript of the document node in the train set.

$$Z = softmax\left(W_e^{(L)} Z^{(L-1)} + b_e^{(L)}\right) \tag{10}$$

$$\mathcal{L}_{DNN} = -\sum_{i \in D_{\mathrm{train}}} \sum_{j=1}^{C} Y_{ij} \cdot log\, Z_{ij} \tag{11}$$

The last layer of the GCN module is a multi-classification layer with a nested softmax function. The loss function of topic classification can be expressed as formula (12) and (13).

$$H^{(L)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \tilde{A}_\tau \cdot \widetilde{H}_\tau^{(L-1)} \cdot W_\tau^{(L-1)}\right) \tag{12}$$

$$\mathcal{L}_{GCN} = -\sum_{i \in D_{train}} \sum_{j=1}^{C} Y_{ij} \cdot log\, H_{ij} \tag{13}$$

The representations learned from both the DNN module and the GCN module can be used to discover document topics, so the final loss includes the prediction loss of both DNN and GCN.

$$\mathcal{L} = \mathcal{L}_{DNN} + \mathcal{L}_{GCN} + \eta \|\Theta\|_2 \tag{14}$$

The loss is shown by formula (14), where Θ denotes the model parameters and η denotes the regularization factor. Through this joint training mechanism, BERT-HGCN can learn two representations simultaneously, one is a text-only representation while the other is a graph-structured representation dominated by multiple semantic associations.

IV. Experimental Results and Discussions

A. Dataset

In order to verify the effectiveness of the proposed topic mining model, we constructed a dataset based on organization-related news data from August 2, 2021, to November 2, 2021. First we filter documents based on duplicate titles. Then, two annotators are required to assign each document with one category among academic conferences, science lectures, inventions and papers, biography, policy recommendations, and others. The Kappa coefficient of the final annotation result is 0.76. The preprocess results in 10k documents. In the modeling stage, a heterogeneous graph is constructed based on word-word, word-document, and document-document relation. The details of the constructed dataset are shown in TABLE I and TABLE II.

TABLE I. Statistics of the Dataset

$\#Documents$	$\#Organizations$	$\#Topics$
10189	102	6
$\#Words$	**Avg. Title Length**	**Avg. Text Length**
19235	25	499
$\#Edges_{Word-Word}$	$\#Edges_{Doc-Word}$	$\#Edges_{Doc-Doc}$
8842094	2052286	107378

TABLE II. Distribution of the Topics

Topic	Percentage(%)
academic conferences	13.43
science lectures	18.28
inventions and papers	25.04
biography	11.68
policy recommendations	6.05
others	25.52

B. Baselines

We compare the proposed model with two types of methods, one considers only document-level semantic information, and the other mainly considers relationships between documents.

1) Type I: Baseline models only consider document-level semantic information via inductive learning

- **TF-IDF-LR** [16]: It adopts the bag-of-words model to obtain the TF-IDF features of the document, and then uses logistic regression as the classifier for topic mining.
- **Bi-LSTM** [3]: It has a forward and backward LSTM, hence both previous and future information is captured to encode the full documents, followed by a linearity classifier to mine topics.

- **BERT** [10]: The method uses pretrained BERT model as encoder with a sentence classification unit on top.
- **RoBERTa** [19]: It uses pretrained RoBERTa model as encoder for the full-document representation, followed by a linearity classifier.

2) Type II : Baseline models mainly consider relationships between documents via transductive learning

- **TextGCN** [6]: Graph Convolutional Networks for Text Classification. It uses nodes to represent documents and words, and the nodes are connected by different kinds of edges. The node representation is learned by a heterogeneous graph convolutional neural network.
- **BertGCN/RoBERTaGCN** [8]: Transductive Text Classification by Combining GNN and BERT. It co-trains the heterogeneous GCN and the pre-trained language model via the memory bank mechanism, which initializes nodes with pre-trained text representations.

To make a fair comparison and perform more comprehensive analysis, we implement several variants of models in Type II enhanced with doc-doc edge, which are marked as TextGCN(KNN) and BERTGCN(KNN).

3) Type III: Models proposed in this work

- **BERT-HGCN**: Model proposed in this work based on BERT, which incorporates word-word-doc-doc relationship and text content for topic mining.
- **RoBERTa-HGCN**: Similar to BERT-HGCN, which considers multi-semantic association and text information based on RoBERTa.

C. Experimental Settings

The constructed dataset is randomly split into training set, validation set, and test set according to the ratio of 7:1:2. For all the methods, the hyperparameters are selected using the valid dataset via grid search. For the methods that use TF-IDF as features, we filter words that appear less than 5 times. For the methods that use PMI, we set the sliding window size as 15 and only keep the positive PMI value. We employ ConSERT[21] to fetch document embeddings and calculate similarity between them, then select top 5 similar documents using faiss[22]. For Bi-LSTM, we use static Chinese word embedding[23] and set the hidden size as 300. For the BERT/RoBERTa based models, we utilize Chinese BERT models from [24] as base encoder and train the model for a maximum number of 50 epochs , which uses Adam with max learning rate 3e-5 and linear decay. For training BERT/RoBERTa-HGCN, the hidden layer vector dimension is set to 200, the number of layers for DNN and HGCN is set to 3, and the balance factor between HGCN and DNN is set to 0.7. Besides, we use dropout strategy with ratio of 0.2.

The topic mining performance is evaluated by test accuracy (denote as Acc) and marco-averaged F1 score (denote as F1) following [25].

D. Experimental Analysis

1) Models comparison

TABLE III reports the performance comparison of our proposed model with baseline models and ablation experiments.

TABLE III. PERFORMANCE OF DIFFERENT METHODS

Type	Methods	F1	Acc
Inductive model	TF-IDF-LR	63.42	68.26
	Bi-LSTM	71.07	72.22
	BERT	76.81	77.67
	RoBERTa	77.54	78.95
Transductive Model	TextGCN	65.19	67.28
	TextGCN(KNN)	64.04	67.67
	BERTGCN	77.91	79.63
	RoBERTaGCN	77.12	79.29
	BERTGCN(KNN)	78.61	79.68
	RoBERTaGCN(KNN)	78.10	79.93
Proposed Model	BERT-HGCN	**80.38**	**81.27**
	RoBERTa-HGNN	79.64	81.18

Comparison among methods only consider document-level semantic information via inductive learning. This kinds of models include four models, namely TF-IDF-LR, Bi-LSTM, BERT, and RoBERTa. We compare these baselines to explore the way to better encode report text. As we can see from TABLE III, among the Type I models, the MacroF1 ranges from 63.42% to 77.54%, the Acc ranges from 68.26% to 78.95%, and RoBERTa performs the best. The pre-trained language model outperforms the TF-IDF and LSTM models, because the pre-trained language model generates better contextual representations with the help of the multi-head attention mechanism in the multi-layer transformer block.

Comparison among methods mainly consider relationships between documents via transductive learning. Among the Type II models, the MacroF1 ranges from 64.04% to 78.61%, the Acc ranges from 67.28% to 79.93%. And BERTGCN(KNN) obtains the highest F1, RoBERTa-GCN(KNN) gets the highest Acc. Based on conduction learning, KNN classifiers perform better than linear classifiers. In addition, the F1 and Accuracy of BERTGCN reached 77.91 and 79.63 respectively, which has also been improved compared to TextGCN. This shows that the pre-trained model plays an important role in topic mining.

All methods compared. Compared with the inductive and transductive models, BERTGCN adds word-document and word-word semantic relationship modeling on the basis of BERT, so the topic performance is further improved. Additionally, we can find that the proposed BERT-HGCN based on heterogeneous graph network and pre-trained language model achieves the best results among all models,

with F1 and Accuracy reaching 80.38 and 81.27, respectively. Compared with BERTGCN, BERT-HGCN uses more types of edge weights, considering that there are order of magnitude differences between different types of edge weights, therefore, different types of nodes should have different information fusion methods. In addition, in terms of the combination of pre-trained language model and graph network representation learning, the degree of integration of BERT-HGCN is deeper than that of BERTGCN, and it can perform better than BERTGCN.

2) Effect of different modules in node representation learning

The balance factor reflects the role of DNN and HGCN modules in topic mining. The experimental results of the balance factor parameter in the loss function in BERT-HGCN are shown in Fig. 2. When other parameters remain unchanged and the balance factor increases to a certain extent, the effect of the BERT-HGCN model is improved. This shows that although HGCN considers more granularity of information, the introduced DNN representation can also alleviate the over-smoothing problem in the learning process of graph network representation to a certain extent.

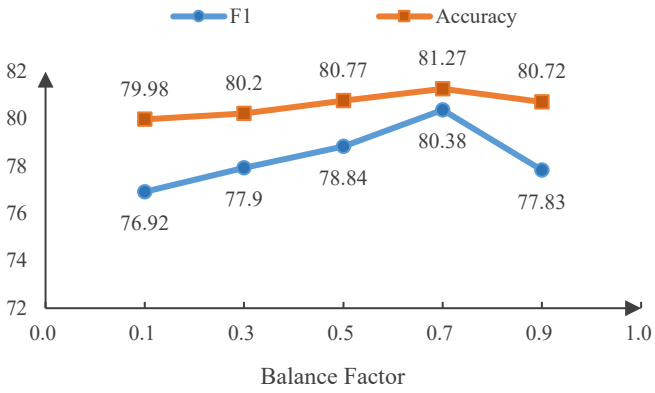

Fig. 2. Effect of the balance factor

3) Effect of parameter sensitivity

a) Effect of vector dimension

Fig. 3 shows the results when the node vector dimension of the BERT-HGCN model changes while other parameters remain unchanged under the topic mining task. As the dimension increases, the performance of the BERT-HGCN model first increases and then decreases. Therefore, choosing an appropriate vector dimension is beneficial to obtain a better text representation.

b) Effect of regularization parameter

In this model, parameter dropout reflects the degree of regularization. As shown in Fig. 4, when the dropout parameter is increased to a certain extent with other parameters unchanged, the performance of the BERT-HGCN model is improved. However, when the dropout parameter continues to increase, the F1 and accuracy achieved by BERT-HGCN decreases. The results show that in the process of text representation learning, a certain proportion of dropout can play a role in regularization, which can prevent overfitting and increase the generalization performance of the model , however, too large dropout ratio will reduce the learning ability of the model.

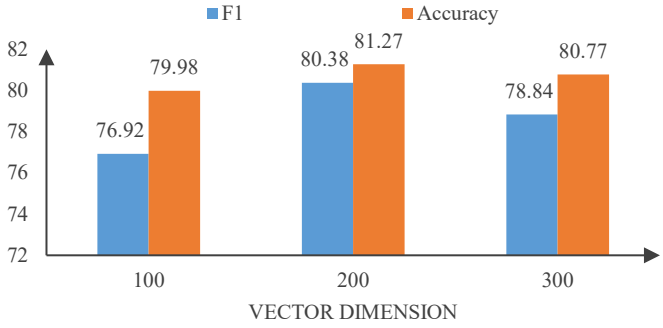

Fig. 3. Effect of the vector dimension

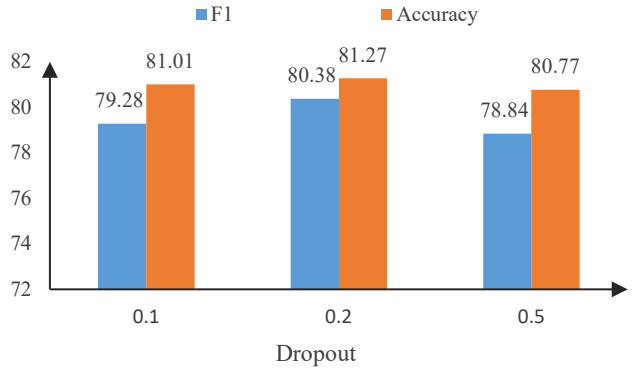

Fig. 4. Effect of the regularization parameter

V. Conclusion and Future Work

In this paper, we propose a BERT-based multi-heterogeneous graph neural network model that comprehensively considers multi-semantic relations between words and documents. It constructs graph that takes words and documents as nodes, regards multiple semantic interactions as edges. During the model learning process, a two-stream encoding mechanism is used to learn the structural and semantic representations, which fuses a hierarchical graph convolution network and a BERT-based DNN auto-encoder. Finally, a dual-supervised loss function is used to train a classifier based on graph nodes and semantic representations for topic mining. Experimental results on real-world dataset demonstrate the effectiveness of the proposed model. In future work, we can integrate other information such as organization's profile for expanding the amount of domain information, thereby improving the performance of topic mining.

VI. ACKNOWLEDGEMENT

This work was partially supported by the National Key Research and Development Program of China (Grant No.2020AAA0103405), the National Natural Science Foundation of China (Grant No.62071467, 71621002), and the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA27030100).

REFERENCES

[1] A. Abbasi, Y. Zhou, S. Deng, and P. Zhang, "Text Analytics to Support Sense-Making in Social Media: A Language-Action Perspective," *MIS Q.,* vol. 42, 2018.

[2] S. Lacoste-Julien, F. Sha, and M. I. Jordan, "DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification," in *NIPS*, 2008.

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, pp. 1735-1780, 1997.

[4] A. Vaswani *et al.*, "Attention is All you Need," *ArXiv,* vol. abs/1706.03762, 2017.

[5] L. Yao, C. Mao, and Y. Luo, "Graph Convolutional Networks for Text Classification," in *AAAI*, 2019.

[6] L. Hu, T. Yang, C. Shi, H. Ji, and X. Li, "Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification," in *EMNLP*, 2019.

[7] Z. Wang, X. Liu, P.-Y. Yang, S. Liu, and Z. Wang, "Cross-lingual Text Classification with Heterogeneous Graph Neural Network," in *ACL/IJCNLP*, 2021.

[8] Y. Lin *et al.*, "BertGCN: Transductive Text Classification by Combining GNN and BERT," in *FINDINGS*, 2021.

[9] Z. Zhu, X. Fan, X. Chu, and J. Bi, "HGCN: A Heterogeneous Graph Convolutional Network-Based Deep Learning Model Toward Collective Classification," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* 2020.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *NAACL-HLT*, 2019.

[11] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR '99*, 1999.

[12] D. M. Blei, A. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.,* vol. 3, pp. 993-1022, 2003.

[13] Y. Meng *et al.*, "Discriminative Topic Mining via Category-Name Guided Text Embedding," *Proceedings of The Web Conference 2020,* 2020.

[14] Y. Meng, Y. Zhang, J. Huang, Y. Zhang, C. Zhang, and J. Han, "Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* 2020.

[15] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Commun. ACM,* vol. 18, pp. 613-620, 1975.

[16] J. E. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," 2003.

[17] T. Kenter, A. Borisov, and M. de Rijke, "Siamese CBOW: Optimizing Word Embeddings for Sentence Representations," *ArXiv,* vol. abs/1606.04640, 2016.

[18] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *EMNLP*, 2014.

[19] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *ArXiv,* vol. abs/1907.11692, 2019.

[20] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks,* vol. 20, pp. 61-80, 2009.

[21] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer," *ArXiv,* vol. abs/2105.11741, 2021.

[22] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs," *IEEE Transactions on Big Data,* vol. 7, pp. 535-547, 2021.

[23] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, "Analogical Reasoning on Chinese Morphological and Semantic Relations," in *ACL*, 2018.

[24] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting Pre-Trained Models for Chinese Natural Language Processing," *ArXiv,* vol. abs/2004.13922, 2020.

[25] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks," *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2015.