Applied Intelligence (2023) 53:20174–20190
https://doi.org/10.1007/s10489-023-04556-x

A multi-semantic passing framework for semi-supervised long text classification

Wei Ai[1] · Ze Wang[1] · Hongen Shao[1] · Tao Meng[1] · Keqin Li[2]

Accepted: 1 March 2023 / Published online: 31 March 2023

Abstract
As an important task of natural language processing (NLP), text classification has flourished with the rise of deep learning techniques. However, existing deep learning methods face challenges as the length of input text increases. Many long text classification works are classified by text truncation or simply extracting keywords, which leads to the loss of rich semantic and structural information. Furthermore, there are great demands for studying semi-supervised long text classification due to the lack of labeled training data and continuously generated long texts in different stylistic. To alleviate these problems, we propose a heterogeneous attention network method based on a multi-semantic passing framework. In particular, we develop a flexible heterogeneous information graph to model the long texts by extracting information, including keywords, entities, titles, and their multi-interrelation. It can effectively integrate the semantic relationship and condense the global information to preserve the significant semantic and structural information well. Furthermore, we design a multi-semantic passing framework capable of extracting the semantic and structural information in the constructed heterogeneous information graph by the semantic degree of specific structures. Experimental works on four real-world datasets are studied, such as ThuCNews, SougouNews, 20NG, and Ohsumed, yielded outstanding results. It is shown an accuracy rate of 98.13%, 98.69%, 87.62%, and 71.46%, respectively, which performs better than the existing methods.

Keywords Graph neural network · Heterogeneous information graph · Long text classification · Semantic information

1 Introduction

With the rapid development of social media, much text data is always generated. Therefore, obtaining adequate information from massive network data has already become a research hotspot in academia. Text classification plays an essential role in information extraction as one of the basic tasks of NLP, which has many applications, including question answering, spam detection, sentiment analysis, news categorization, user intent classification, etc [16]. In recent years, many deep learning methods have been proposed to promote the development of text classification research [2, 11, 23, 28, 31]. However, most of these existing deep-learning methods have several challenges as the length of the input text increases.

Recently, long text classification has been nontrivial due to the following challenges. The first challenge is that it is difficult to preserve and extract useful information from long texts after they have been preprocessed due to their prosperous and complex information. The most direct and easiest method to solve this problem is to process the long text into multiple short pieces and process them separately. It is contained the following two types of processing methods. The first type is to truncate a specific character length in order. One of the most well-known methods is a transformer-based pre-train model named Bert, which is used a masked language model and limits input length to pre-train the bidirectional transformers [3]. The second type is to select specific paragraphs or sentences to represent the text. For example, Chen et al. [1] constructed a multi-task architecture, which jointly trains an Albert [12] model to key-sentence extraction with distance square loss and multi-label long text classification tasks with cross-entropy loss. To better capture the semantics of long texts, Du et al. [5] proposed a Knowledge-Aware Leap-LSTM to skip irrelevant words in the input for accelerating LSTM models by integrating prior human knowledge. However, this

✉ Tao Meng
mengtao@hnu.edu.cn

Extended author information available on the last page of the article.

method of inputting the entire text sequence for processing can not distinguish the noise information. Therefore, it can not accurately extract important information from the text. Although the methods of truncation and selection can condense long texts to some extent, there still inevitably ignore part of the semantic and structural information, which leads to the loss of essential information and results in misjudgment of the model.

The second challenge is the complicated construction of the training set of long text. Massive new and different stylistic text data are constantly generated, which requires a lot of new labeled data for existing deep learning models to learn. The emergence of graph convolutional neural networks (GNN) provided a new direction for solving the problem [10]. It aggregated the information of neighbor nodes in the relational network constructed by different texts to achieve similar effects as other methods, but only required a small part of the labeled data. Meanwhile, GNN-based models can better preserve the structural and semantic information by modeling the corpus. For example, Yao et al. [30] built a text graph for corpus based on word co-occurrences and document-word relationships, then jointly learned the embeddings for both words and documents by graph convolutional neural networks. Ragesh et al. [20] designed a heterogeneous graph convolutional network modeling approach to learn feature embeddings and derive document embeddings by combining the best aspects of PTE [22] and TextGCN [30]. Moreover, Linmei et al. [13] proposed a heterogeneous graph attention network with two-level attention mechanisms for learning the importance of different neighboring nodes and node types to a current node. These GNN-based models can aggregate the information of neighbors to strengthen the representation of nodes by semi-supervised learning. After these existing methods simply construct a heterogeneous graph through documents or keywords, the authors believe that the most important thing is to enrich the representation of the node itself through the neighbors. However, it is necessary to consider the semantic relationship in the text and the high-order semantic structure in the graph, which is because the long text contains too many words and complex features. Unfortunately, these methods do not take these aspects into account.

To address the above problems, we propose a novel **H**eterogeneous **A**ttention **N**etwork for semi-supervised **L**ong **T**ext classification (Han-LT). Firstly, according to the characteristics of long text, the definition of multi-interrelation based on entity-keyword-title is defined. We extract the titles, entities, and keywords from the texts and get their initial embeddings. Then, their multi-interrelation is found within and between texts, building edges by the multi-interrelation to construct the heterogeneous information graph. In this way, the semantic and structural information of long texts can be preserved to a great extent. Secondly, the multi-semantic passing framework is designed to extract crucial semantic and structural information. Specifically, we first put forward the definition of the semantic degree to measure the importance of different semantic structures in the heterogeneous information graph. Then, the attention mechanism and the semantic degree are combined to capture high-order semantic information while capturing the importance difference of neighbor nodes. Finally, we construct a heterogeneous neural network named Han-LT based on the multi-interrelation heterogeneous information graph and the multi-semantic passing framework to get the classification results by adding the softmax layer at the end of the network. The main contributions of this paper can be summarized as follows:

- We construct a novel heterogeneous information graph for long texts by extracting titles, entities, keywords, and their multi-interrelation to preserve their significant semantic and structural information.
- We design a special multi-semantic passing framework for capturing the importance of different nodes, higher-order semantics, and structural information by combing the attention mechanism and the semantic degree.
- We evaluate the effects of the Han-LT and compare it with 7 state-of-the-art methods. Extensive experimental results show the superiority of our Han-LT method on the long text classification task.

In Section 2, we will introduce the related work of this paper from text classification in deep learning and graph neural networks. In Section 3, we will elaborate on our Han-LT method. In Section 4, we present a large number of designed experiments, experimental results, and related analysis verify the superiority of Han-LT. The Section 5 is the conclusion of this paper.

2 Related work

2.1 Text classification in deep learning

In the past decades, text classification has gradually changed from a shallow learning model to a deep learning model. Deep learning methods can avoid the manual design of rules and functions. The proposal of convolutional neural networks (CNN) [11] was aimed at image classification, which has achieved subversive achievements and promoted the arrival of the hot era of deep learning. In order to apply CNN to text classification tasks, Kim et al. put forward a convolutional neural network called TextCNN [9]. It took an embedding obtained with a pre-trained word vector method as input, which determined the discriminative phrase through one convolution layer and one max-pooling layer. Tan et al. [21] utilized gated units and shortcut connections to transform and carry word information to control how much context information is incorporated into each specific position of the word embedding matrix in the text. Considering long-range or sequential semantics, Peng et al. [18] inputted the word matrix that maintained word order into attention graph capsule recursive CNN to learn semantic features, then a hierarchical classification embedding method was designed to learn the hierarchical relationship between category labels. To alleviate computational complexity, Johnson et al. [8] developed a low-complexity, word-level deep convolutional neural network for text classification called DPCNN. It could obtain a global representation of text by deepening the network without greatly increasing the computational cost. However, the effect of applying them directly to long texts is not satisfactory. The excessive length of the long text can make the graph too complex, which results in the disappearance of gradient or network degradation.

Recurrent Neural Networks (RNNs) have been widely used to capture long-term dependencies through recursive computation, and the performance in long text classification tasks is better than CNNs. For instance, Liu et al. [14] designed a model to capture long text semantics, which could extract context information and effectively reduce the time complexity of the model. Du et al. [4] proposed the Pointer-LSTM framework, which relied on a pointer network to select important words for target prediction. It generated self-attention distribution over the whole input sequence through a small bidirectional LSTM network. Then, a large BiLSTM network was used to obtain Top-k keywords for target prediction. Later, the authors put forward the Knowledge-Aware Leap-LSTM [5] to skip irrelevant words in the input for accelerating RNN models by integrating prior human knowledge. It integrated prior knowledge through factorized and gated integration to partially supervised the word-skipping process, which achieved higher accuracy and faster training speed. Moreover, Du et al. [6] proposed recurrent BLS (R-BLS) and long short-term memory (LSTM) architecture: gated BLS (G-BLS) to learn multiple information simultaneously to achieve high accuracy in text classification. Unfortunately, the gradient problem of those RNN-based methods still exists and might be intractable when facing longer sequences. In addition, all those CNN-based and RNN-based methods were data-driven, which usually required a large amount of high-quality labeled data or prior professional knowledge to achieve higher performance.

The emergence of pre-training models, such as Bert [3], GPT [19], XLNet [29], MacBERT [26], etc., has greatly promoted the development of text classification, especially for long and ultra-long texts. Bert adopted a novel masked language model to pre-train bidirectional transformers to generate deep bidirectional language representations. After pre-training, it was necessary to add an output layer for fine-tuning to achieve state-of-the-art performance in tasks such as text classification. Meanwhile, unsupervised learning under large-scale data has significantly improved the classification effect of the model. However, the mechanism of Bert required the text to be truncated. It was shown that part of the semantic and global information was missing, which made the classification results more likely to be disturbed by noise.

2.2 GNN for text classification

The appearance of GNN provided a new idea for the text classification task, which was transformed into a graph node classification task. GNN-based text classification methods could capture the structural information of texts, and other methods can not replace it.

In recent years, Graph Convolutional Networks (GCN) [10] performed convolution operations on graph structure data and achieved attractive performance in various tasks. They could encode the characteristics of the graph structure and nodes without designing features or fusion methods. Many variants of GCNs were proposed over the next few years. These methods could be divided into 1) homogeneous graph neural network and 2) heterogeneous graph neural network. The difference between the two methods lies in how the graph was constructed and processed.

GraphSAGE [7] was a classic algorithm based on airspace. It improved the traditional GCN in two aspects. During training, the sampling method optimized the full-graph sampling of GCN to partial node-centered neighbor sampling. The second aspect was that GraphSAGE studied several ways of neighbor aggregation. GAT [24] aggregated neighbor nodes through a self-attention mechanism to achieve adaptive matching of the weights of different neighbors, which improved the accuracy of the model. Moreover, Yao et al. [30] designed a text graph convolutional network (TextGCN), which constructed a heterogeneous word text graph for the entire data set and captured global word co-occurrence information. These homogeneous graph neural network methods have achieved remarkable results in multiple fields. However, most networks in reality are heterogeneous. It is essential to build and deal with heterogeneous graphs according to the actual situation. Zhang et al. [32] constructed the HetGNN model for processing the heterogeneous graphs, which used LSTM for node-level aggregation and an attention mechanism for semantic-level aggregation. It could simultaneously capture the heterogeneity of structure and content, which is suitable for transductive and inductive tasks. Heterogeneous Graph Attention Network (HGAT) with a two-level attention mechanism could learn the importance of different adjacent nodes and node types in the current node [27]. It propagated information on the graph and captured relationships to solve the semantic sparsity problem of semi-supervised short text classification. Ragesh et al. [20] designed a heterogeneous graph convolutional network modeling approach which utilized across layers to learn feature embeddings and derive document embeddings. It greatly reduced the model's parameters and achieved better performance.

These GNN-based models have made remarkable achievements in text classification tasks by aggregating the information of node's neighbors to enrich the embedding about the node itself. However, most of these methods used chapter-level texts as nodes or simply extracted keywords as text embeddings, which inevitably led to excessive computation or loss of semantic information if applied to the long text classification task.

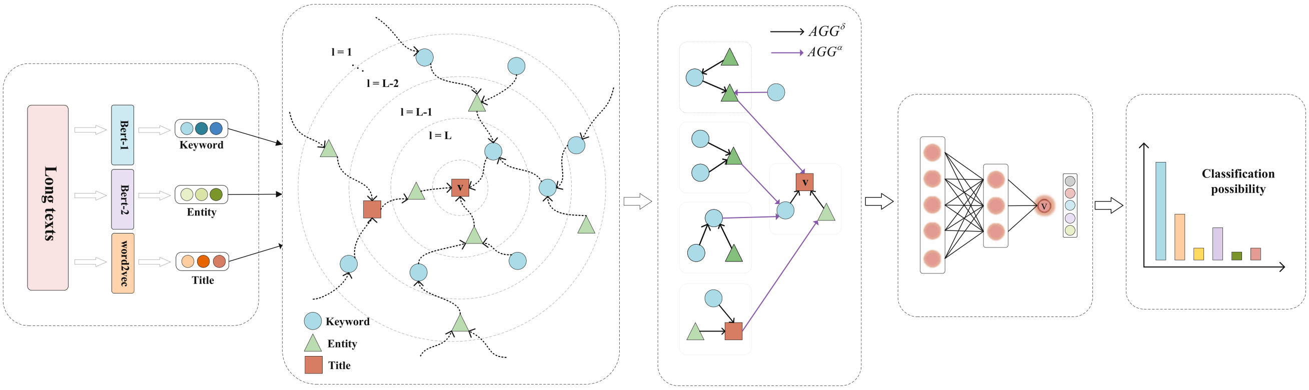
3 The proposed method

In this paper, we propose a novel semi-supervised long-text classification method named Han-LT, which can take advantage of limited labeled data to preserve and extract the significant structural and semantic information. The general process of Han-LT is shown in Fig. 1. Firstly, we extract titles, entities, and keywords from long texts and get their initial embeddings by using Bert and Word2vec [15]. Secondly, we give the definition of multi-interrelation based on the entity-keyword-title. The heterogeneous information graph is built based on the multi-interrelation to preserve the long texts' semantic and structural information. Thirdly, the definition of the semantic degree is used to measure the importance of different semantic structures in the heterogeneous information graph. By combining the semantic degree and attention mechanism, we design the multi-semantic passing framework to capture the relationship of nodes and extract the higher-order semantic and structural information. Finally, a softmax layer is added at the end of the network to obtain the final classification results.

3.1 Multi-interrelation heterogeneous information graph

Due to the high complexity of features, the tasks of long text classification face many challenges. The most difficult one is extracting valuable and essential information from complex features. Existing graph neural network methods usually construct information graphs simply from documents or keywords. This approach does not consider retaining semantic information from the internal level of the text, which results in the loss of the key information in subsequent processing. To address this issue, we present a heterogeneous graph construction method for long text classification task. Specifically, we put forward the definition of multi-interrelation based on the entity-keyword-title to preserve the core semantic and structural information in long texts. The graph construction method is mainly divided into two steps. Firstly, we extract the titles, entities, and keywords from the texts and get their initial embeddings. Secondly, the multi-interrelation within and between texts are found and built edges for them according to the multi-interrelation.

Here, we consider constructing the heterogeneous information graph $G = (V, \xi)$ including entities



(a) Information extraction (b) Heterogeneous information graph (c) Multi-semantic passing (d) Graph convolutional encode (e) Label classification

Fig. 1 Illustration of our method Han-LT. Among them, (a) represents the acquisition of keyword, entities, titles, and their initial embedding. (b) represents the heterogeneous information graph constructed by the multi-interrelation. In (c) multi-semantic passing framework, AGG^{δ} represents the multi-semantic passing mecanism, AGG^{α} represents the attention mechanism. (d) represents the graph convolution layers and (e) represents the final classification result

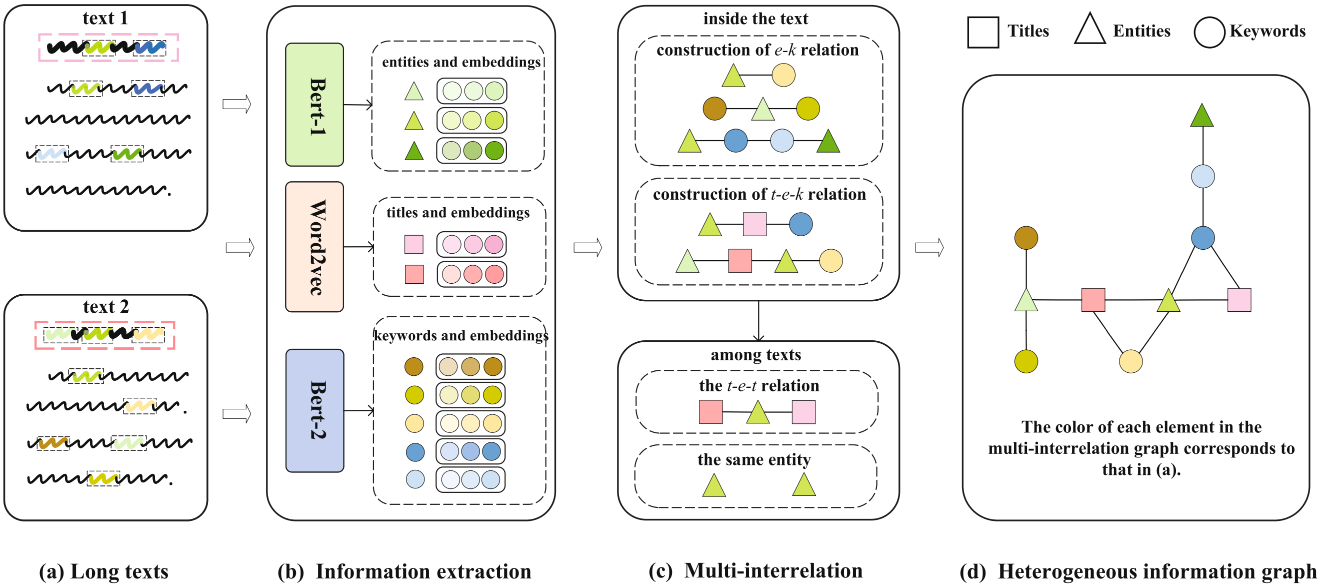


Fig. 2 Illustration of the multi-interrelation heterogeneous information graph for long texts

$E = \{e_1, ..., e_m\}$, keywords $K = \{k_1, ..., k_s\}$, titles $T = \{t_1, ..., t_n\}$, and $V = E \cup K \cup T$. ξ represents the relationship between nodes. The details of the graph construction are shown in Fig. 2, which are described in the next paragraphs.

3.1.1 Information extraction

The title information is extracted directly as the title has been placed in the first row in most cases. Then the trained model Berts are used to extract keywords and entities. Compared to other methods, Bert introduced Masked Language Model (MLM) and Next Sentence Prediction (NSP) in pre-training to train bidirectional features and capture the connection between two sentences. Therefore, the model has the ability to understand the connection of long sequence contexts. Furthermore, a large-scale unlabeled corpus is used for pre-training, so that the model contains text representation information with rich semantics. The text is processed into three embeddings (B_w, B_s, B_p) and used as input to Bert. B_w is the word embedding. B_s is the segment embedding to help Bert distinguish between paired input sequences. B_p is the position embedding, which indicates the index embedding of the position of the current word. The three embeddings are summed with dimensions (1, n, 768) to get the final B_{input} as Bert's input, where n represents the number of words in the text. At the end of the model, a fully connected layer is followed to obtain a 256-dimensional word embedding. Then, it is fine-tuned through the labeled keywords and entity corpus to make it have qualified extraction ability.

The embedding obtained by Berts is used as the initialization embedding of keywords and entities. For the titles, the Word2vec is chosen to embed them. It is worth noting that we treat the title as a separate sentence containing the core intent of the article, so the title and article information need to be separately processed when considering the relationship between texts. The semantics of general titles are relatively complete, and the words in the title can well represent their semantics. Furthermore, taking the efficiency factor into consideration, Word2vec is finally chosen to embed the titles.

3.1.2 Multi-interrelation and edge construction

The construction of edges between different nodes is completed according to the defined multi-interrelation and position information. Specifically, we construct a corresponding sub-graph for each text according to the multi-interrelation between different nodes. Then, the connection between texts through the titles and entities is realized to obtain the multi-interrelation heterogeneous information graph finally.

Multi-interrelation: Inside the text, the interrelation is expressed as the relationship of e_m-k_s in each sentence and the relationship of t_n-e_m-k_s in the title, where k_s and e_m appear in t_n. Among texts, interrelation is expressed as the relationship t_i-t_j or t_i-e_m-t_j, and the relationship between the same entities appearing in different texts and their interactive information.

The relationship between entities and keywords (e_m-k_s) can represent the specific intent of the text, while the relationship of title-entity-keywords (t_n-e_m-k_s) can represent the core intent of the text. The relationship of title-title (t_i-t_j) and title-entity-title (t_i-e_m-t_j) can connect similar titles. As a core element of an article, a specific entity often appears in certain types of texts. Therefore, entities themselves have rich features and strong characteristics. We connect texts that contain the same entity.

Inside the text, we construct edges through the multi-interrelation between entities and keywords in each sentence. Entities and keywords in the same sentence are connected in the order of their appearance to complete the construction of the e_m-k_s relationship. Among texts, we construct the relationship between texts through the entities and the titles. Considering the relationship between texts, we regard the title as an independent sentence that contains the core intent of the article. If two titles contain the same entity, they will be connected through this entity to complete the construction of t_i-e_m-t_j. Moreover, articles with similar titles are more likely to belong to the same category. Therefore we set similarity score s to measure the similarity of two titles if they do not contain the same entity. The similarity score s between title t_i and title t_j can be formulate as follows:

$$s = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\left(\sum_{i=1}^{n} A_i^2\right)^{\frac{1}{2}} \cdot \left(\sum_{i=1}^{n} B_i^2\right)^{\frac{1}{2}}}, \tag{1}$$

where A and B represent the vectors of t_i and t_j, respectively. And n represents the dimension of the vector. If the similarity score s between the title t_i and t_j is greater than the set threshold, the t_i-t_i relationship will be constructed successfully. As for entities, we regard multiple identical entities in different texts as the same node. The same keywords appearing in different texts are regarded as different nodes. The reason is that the same entity represents the same semantics in different articles in most cases, while keywords do not. Therefore, we connect different texts through entities and titles, while keywords are connected with their corresponding entities and titles. In this way, different texts can be related by title and entity information and keep their established multi-interrelation. Furthermore, in Fig. 2(d), the color of each element in the heterogeneous information graph is one-to-one corresponding to that in Fig. 2(a). It can help us better understand the multi-interrelation heterogeneous information graph construction method.

The essential semantic and structural information of long texts are well preserved by constructing the novel multi-interrelation heterogeneous information graph. It reduces much redundant information, which greatly benefits the subsequent classification tasks.

3.2 Multi-semantic passing framework

How to extract and represent the key information in a heterogeneous information graph is a complex problem in graph neural network-based long text classification tasks. However, existing methods are more concerned with enriching the representation of the node itself through the neighbors. It inevitably loses important high-order semantic and local structural information, especially for complex information bodies such as long texts. To further capture the significant information, we design a novel multi-semantic passing framework based on the definition of semantic degree. It can aggregate the information about surrounding different types of neighbors to obtain higher-order semantic information. Especially combined with the constructed multi-relationship heterogeneous information graph containing title, entity, and keyword information, better relevant information can be extracted.

Semantic degree: The proportion of each specific semantic structure among all semantic structures in the multi-interrelation heterogeneous information graph.

The process is described as follows. Firstly, we search and extract specific semantic structures in the heterogeneous information graph according to MotifNet [17], which analyzes integrated networks and searches for specific structures. Secondly, we define the semantic degree to measure the importance of specific semantic structures. With the combination of the semantic degree and attention mechanism, the mutual importance between different nodes can be obtained. The high-order semantic information can be captured according to the semantic degree of the structure where the nodes are located. Finally, the semantic information retained by the node and the neighbors' information is locally propagated. The illustration of our multi-semantic passing framework is shown in Fig. 3.

3.2.1 Semantic degree of edge

In other networks, the motif is a metric used to measure the significance of a structure in a graph. In the heterogeneous information graph of long texts constructed on the entity-keyword-title, motifs represent the core semantic information of texts to a large extent. For example, the entity can be the subject of an event in the structure of entity-keyword-keyword. The keyword can be time, action, or a certain noun or adjective. Then, such a semantic structure can contain rich and important semantic information. Based on the semantic degree, we assign a weight to each edge that is in a specific semantic structure. Formally, the semantic

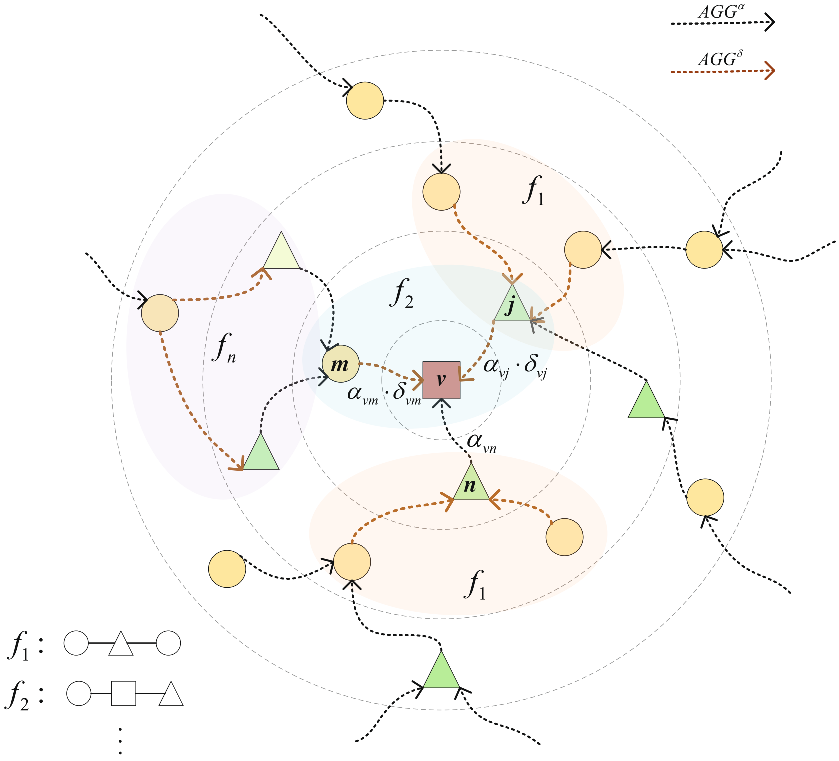


Fig. 3 Illustration of our multi-semantic passing framework

structure is denoted as $f \in F, F = (f_1, f_2, ..., f_k)$, k means all categories of semantic structures. The semantic degree ρ_f is given by

$$\rho_f = 1 + \left(\frac{X_f}{\sum_{f \in F} X_f}\right)^{\frac{1}{2}}, \tag{2}$$

where ρ_f represents the semantic degree of each edge under the structure f. X_f represents the number of structures occupied in the entire heterogeneous information graph. The corresponding weights are set for each edge based on the semantic degree to realize the difference between the feature vectors of different nodes in aggregation. Besides, it is worth noting that some edges are not in any particular semantic structure, while some may be in multiple semantic structures. To accurately measure the semantic weight contained in each edge, the semantic degree calculation formula of each edge is defined as

$$\delta_{ij} = \prod_{e_{ij}\_in\_f} (\rho_f), \tag{3}$$

where δ_{ij} represents the product of the semantic degrees of all semantic structures where the edge e_{ij} is located, namely the semantic degree of the edge. The more types of semantic structures an edge is located in and the higher the semantic degree of the semantic structures, and the larger the value of δ_{ij} are.

3.2.2 Multi-semantic message passing

According to the obtained semantic degree, a multi-semantic passing framework is designed for extracting the important higher-order semantics of long texts. Formally, for a graph $G = (V, \xi)$, let $X \in R^{m*n}$ be the feature matrix of the nodes, where each row is the feature vector of node v. A is the adjacency matrix of G and D is the degree matrix, where $D_{ii} = \sum_j A_{ij}$. Moreover, each node is connected to itself. Then, according to the aggregation function AGG, the neighbors' information of node v is aggregated into N_v to update the embedding of node v recursively. Equations (3) and (4) demonstrate the steps of the attention mechanism:

$$H^l_{N_v} = AGG(H^l_j, v_j \in N_{v_i}), \tag{4}$$

$$H^{l+1}_i = \sigma(\alpha_{ij} \cdot \widetilde{A} \cdot W^l \cdot (H^l_i \oplus H^l_{N_v})), \tag{5}$$

where $\widetilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is the symmetric normalized adjacency matrix. α_{ij} is the attention value of nodes v_i and v_j. It need to be learned by the model, which represents the different importance of each neighbor node to v_i. The operator $\oplus$ denotes concatenation. σ denotes the activation function, such as Leaky ReLU. W^l is the trainable transformation matrix of the layer l. Furthermore, $H_i^0 = X_{v_i}$. The calculation of α_{ij} is shown in Eq. (6),

$$\alpha_{ij} = \frac{\exp(\sigma(\mu^T \cdot [H_i^l \oplus H_i^{l-1}]))}{\sum_{j \in N_{(v_j)}} exp(\sigma(\mu^T \cdot [H_i^l \oplus H_j^{l-1}]))}, \qquad (6)$$

where μ is the attention parameter. $T = (\tau_1, \tau_2, \tau_3)$ are different types of nodes, where τ_1, τ_2, τ_3 represent title, entity, and keyword types respectively. It is worth noting that attention values exist between all nodes, but not all nodes exist in a specific semantic structure. If more than one node in a node pair does not belong to any particular semantic structure, the semantic degree value of the edge between them will be treated as 1. The overall flow of the multi-semantic passing can be expressed as follows:

$$H_i^{l+1} = \sigma(\delta_{ij} \cdot \alpha_{ij} \cdot \widetilde{A} \cdot W^l \cdot (H_i^l \oplus H_{N_v}^l)). \qquad (7)$$

Considering the heterogeneity of different types of nodes, traditional methods generally concatenate the feature spaces of different types of nodes to construct a new large feature space and set the values of other types of irrelevant dimensions to 0 for summation. The obvious disadvantage of is that it ignored the heterogeneous information of different nodes and increased the difficulty of calculation. To optimize this problem, we project different types of nodes into a common space through each type-specific transformation matrix W_τ. Thus, the representation H^{l+1} is given by

$$H^{l+1} = \sigma\left(\sum_{\tau \in T} \delta \cdot \alpha \cdot \widetilde{A}_\tau \cdot W_\tau^l \cdot H_\tau^l\right), \qquad (8)$$

where τ represents the type of neighbor node. The rows of matrix $\widetilde{A}_\tau$ represent all nodes, and the columns represent neighbor nodes of type τ. Then, the neighbor nodes of different types τ are aggregated with different transformation matrices W_τ^l to obtain the final representation H^{l+1} of node v_i. The final aggregation formula can be described as

$$H_i^{l+1} = \sigma\left(\sum_{\tau \in T} \sum_{j \in N_{\tau(i)}} \delta_{ij} \cdot \alpha_{ij} \cdot \widetilde{A}_\tau \cdot W_\tau^l \cdot H_j^l\right), \qquad (9)$$

where $N_{\tau(i)}$ means the set of neighbors of node i belonging to type τ. In this way, the titles, entities, keywords, and the multi-interrelation information between them in the multi-interrelation heterogeneous information graph can be effectively aggregated, which obtains higher-order semantic and structural information.

3.2.3 Label classification

After going through an L-layer Han-LT, we feed the obtained final embedding Q of the long text into a softmax layer for classification. Formally,

$$Z_i = softmax(Q_i^{(L)}). \qquad (10)$$

Moreover, the binary cross-entropy loss function we utilized is as follows,

$$\zeta = -\sum_{i=1}^{N} \sum_{j=1}^{t} (Y_{ij} \log(Z_{ij}) + (1 - Y_{ij}) \log(1 - Z_{ij})), \quad (11)$$

where t is the number of classes, and N is the number of training examples. Y_{ij} denotes the binary ground truth label value, and Z represents the predicted value of the long text i obtained by the Han-LT model, which represents the likelihood that text i will be labeled j.

4 Experiments

Experimental works have been conducted on four common datasets to evaluate the performance of the Han-LT method. This section introduces the data sets and preprocessing, comparison of methods, experiment settings and details, experimental results, and corresponding analysis.

4.1 Datasets and preprocessing

We compare Han-LT with several state-of-the-art methods in different scenarios. Two Chinese datasets and two English datasets are selected from news topic classification and medical disease classification to perform our experiments, they are:

ThuCNews: The ThuCNews corpus is a news document generated by filtering the historical data of the Sina News RSS subscription channel from 2005 to 2011, which contains 14 news categories and about 740,000 news texts. About 6,000 pieces of text data with more than 300 characters are randomly selected for each category.

Sogou News: Sogou News corpus is a news dataset provided by Sogou Lab, including Sogou CA and Sogou CS datasets. It contains about 27,000 news items in ten categories. To balance the dataset, about 3,000 samples were randomly selected for each category, and the number of characters in each sample was greater than 300.

20NG: The 20newsgroups dataset is one of the international standard datasets for text classification, text mining, and information retrieval research. It contains 18,846 non-repeating news texts divided equally into 20 categories.

Table 1 Summary statistics of datasets

Dataset	Docs	Train Docs	Test Docs	Classes	Avg.Length
ThuCNews	84,000	58,800	25,200	14	539.75
Sougou News	30,000	21,000	9,000	10	502.4
20 NG	18,846	13,192	5,654	20	221.26
Ohsumed	7,400	5,180	2,220	23	135.82

Ohsumed: Ohsumed contains 7,400 articles. Each article is a medical abstract with at least one or more labels from 23 cardiovascular disease categories. Since a document may be labeled with multiple labels, the label with the highest level is taken as its final label in the experiments.

The two Chinese datasets are filtered by length to construct two real long text datasets, which makes our results more convincing in long text classification tasks. For all selected datasets, we remove stop words and low-frequency words (word frequencies below 5). We select 70% of each dataset as the training set and the rest as the test set. About 30% of the data in the training set is labeled data. In our datasets, all long texts contain entities that we have defined. The statistics of the pre-processed datasets are detailed in Table 1.

4.2 Comparison of methods

To comprehensively evaluate our method, we compare it with the following 7 state-of-the-art algorithms:

CNN [11]: CNN is a classical neural network that utilizes convolutional computation. We explore a 13-layer CNN with two variants: 1) CNN-rand, which uses randomly initialized word embeddings, and 2) CNN-pre, which uses pre-trained word embeddings.

Bert [3]: It is a pre-trained model that stacks multiple transformer models and pre-trains bidirectional deep representations by conditioning the bidirectional transformers in all layers. We choose an existing trained Bert-base model and fine-tune it to convergence with our training data.

Pointer-LSTM [4]: A LSTM framework that relies on pointer networks to select important words for target prediction. It maintains a consistent input process for the LSTM module and allows it vary the skip rate during inference.

TextGCN [30]: It is used a graph convolutional network to model the corpus for capturing neighborhood information, and it is built a bipartite graph using word co-occurrence information and word frequency information. It is transformed the text classification problem into a node classification problem.

GAT [24]: Graph Attention Networks adopt the attention mechanism to learn the weights of neighbor nodes adaptively. The nodes' expression is obtained through the weighted summation of neighbor nodes.

HAN [25]: It puts forward a novel dual-level attention mechanism, including node-level attention and semantic-level attention. The node-level attention is used to learn the importance between the central node and its different types of neighbor nodes, and the semantic-level attention is used to learn the importance of different meta-paths.

HeteGCN [20]: A heterogeneous graph convolutional network combines the best aspects of PTE and TextGCN. It learns feature embeddings and derives document embeddings using a HeteGCN architecture with different graphs used across layers.

4.3 Experiments settings and details

The following experiments are conducted to compare and analyze our Han-LT method comprehensively. The first experiment provides an overall evaluation of all methods. Our method achieves excellent results on multiple datasets, which demonstrates the effectiveness of Han-LT. The second and third experiments are designed to embody the superiority and flexibility of the multi-interrelation heterogeneous information graph construction method and the multi-semantic passing framework. In the second experiment, the corpus is modeled using Han-LT for constructing a heterogeneous information graph of long texts and then processing them with different heterogeneous graph neural networks. The third experiment is designed to demonstrate the scalability of the multi-semantic passing framework. The framework achieves good results on different heterogeneous graphs based on text classification. The fourth experiment is to demonstrate the superiority of Han-LT in semi-supervised algorithms by changing the proportion of labeled data in the training set. To further verify the superiority of Han-LT on semi-supervised learning, we design the fifth experiment to find out which part of Han-LT has a greater impact on semi-supervised learning. Then, the sixth experiment is designed to demonstrate the selection process of some core parameters in the method. Besides, for the viewability of the experimental tables, the heterogeneous information graph is defined as HIN.

We evaluated the performance of all classifier models using *Acc* and F1 score. *Acc* represents accuracy which represents the correct prediction ratio among all the predicted samples. The $F1$ value is introduced to evaluate the model more comprehensively, which is formulated as

$$F1 = 2 * \frac{P \cdot R}{P + R}, \tag{12}$$

where P represents precision and R represents recall. During model training, we train all models until the loss value converges and repeat this process ten times. Then, average the best accuracy and $F1$ score obtained in each experiment as the final results.

After experimental verification, we select the optimal number of entities and keywords for each document, and the value of title similarity x. To construct the heterogeneous information graph of long texts, we set the maximum number of entities extracted per document $K = 5$, and the maximum number of keywords $J = 10$. For all text corpora, the title similarity threshold x is set to 0.6. Furthermore, the initial dimension of words is set to 258. As for model training, we set the learning rate $l = 0.005$, regularization factor n = 1e-6, and dropout rate as 0.5. All methods are run on a computer with an i7-9700kf CPU and an RTX2070s GPU.

4.4 Experimental results

4.4.1 The overall experiment

Table 2 shows the classification accuracy rate and F1 score of different algorithms on the four datasets. The accuracy rate of Han-LT is higher than 98% on both Chinese long text datasets, of which 98.86% is achieved on Sougou News and 87.62% accuracy on the classic news classification dataset 20NG and 71.46% on the disease classification dataset Ohsumed. Compared with the classical attention mechanism-based method GAT, Han-LT has higher effects on all datasets, and the highest improvement can reach 8.9%. We note that Han-LT is improved by 0.38%–1.56% compared with the new baseline methods (except CNN-based methods) in the Chinese datasets, while Han-LT has an improvement of 0.55%–8.81% compared with the new baseline methods in the English datasets. Compared with the baseline method, the improvement of Han-LT on the English datasets is significantly greater than that on the Chinese datasets. It is because the Chinese datasets have less room for improvement (the accuracy rate of the baseline methods is about 97%). However, Han-LT further improves the effect due to its ability to extract deeper semantic information. In general, it is obtained that Han-LT performs best on all datasets, which represents the Han-LT method's effectiveness and superiority on long text classification tasks.

It is noted that all methods, including Han-LT, outperform the English datasets on the Chinese datasets. After analysis, it is concluded that because the data characteristics of each category in the Chinese datasets are obvious, the data of different categories are quite different. The English data sets do not have the property, especially for Ohsumed, whose original data set may contain multiple labels for each data. It is indicated that their textual information is intricate. Secondly, the English datasets contain fewer entities that we have defined. For example, most of the entities in Ohsumed are medical-related professional vocabulary. However, the proposed Han-LT method can still achieve better results than other methods in such cases.

For more in-depth performance analysis, we note that there are also specific differences within the baseline methods. For instance, the CNN-pre using pre-trained word vectors is significantly improved compared to the CNN-rand which randomly initializes word vectors. It shows the importance of node representation learning. The pretrained model Bert outperforms CNN-pre on two datasets with longer text lengths but is not as good as CNN-pre on 20NG. It is analyzed that CNN can better simulate continuous and short-range semantics while Bert can better capture long-range semantic information. Similarly, the LSTM-based method Pointer-LSTM has a greater advantage in long sequence classification, which performs better on longer texts. Graph neural network-based model TextGCN achieves comparable results with the pret-rained deep model Bert. Compared to the CNN-pre method, the GNN-based methods (such as TextGCN, GAT, HAN, and HeteGCN) can improve up to 9.92% on the Ohsumed dataset is a significant improvement. The overall performance of GAT is better than that of TextGCN on most datasets since the attention mechanism can adaptively learn the weights

Table 2 Test accuracy and F1 score of different methods on two Chinese datasets and two English datasets

Dataset		CNN-rand	CNN-pre	Bert	Point-LSTM	TextGCN	GAT	HAN	HeteGCN	Han-LT
ThuCNews	*Acc*	0.9273	0.9554	0.9677	0.9783	0.9682	0.9754	0.9779	0.9795	**0.9833**
	*F*1	0.924	0.9536	0.9641	0.9745	0.966	0.9728	0.9751	0.9769	**0.9782**
SougouNews	*Acc*	0.9364	0.9581	0.9722	0.9817	0.9734	0.9784	0.9806	0.983	**0.9869**
	*F*1	0.9325	0.9542	0.9694	0.9789	0.9702	0.9733	0.9752	0.9804	**0.9821**
20NG	*Acc*	0.7678	0.8216	0.7923	0.8328	0.8569	0.8619	0.8645	0.8707	**0.8762**
	*F*1	0.7643	0.8197	0.7902	0.8304	0.8515	0.8573	0.8608	0.8665	**0.8714**
Ohsumed	*Acc*	0.4387	0.5844	0.6745	0.6853	0.6836	0.6256	0.632	0.6574	**0.7146**
	*F*1	0.4348	0.5781	0.6687	0.6809	0.6792	0.6194	0.6267	0.6513	**0.7083**

The bold entries shows the best results of the experiments to better demonstrate the effect of Han-LT of neighbor nodes. It is indicated the superiority of the attention mechanism. HeteGCN combines the advantages of TextGCN and PTE to construct a text corpus as a heterogeneous graph, which achieves good results on four datasets. However, these methods do not profoundly consider the semantic information inside the text, resulting in the partial loss of the rich semantic information in the long text. The proposed Han-LT method takes into account and achieves better results.

4.4.2 The analysis of the multi-interrelation heterogeneous information graph

The following experiments are designed to verify the superiority of the constructed heterogeneous information graph for long texts. We apply GAT, GCN, HeteGCN, and HAN to our constructed heterogeneous information graph for classification. GAT and GCN do not consider the heterogeneity of nodes, while HeteGCN and HAN consider the heterogeneity of nodes. In response to this problem, we treat nodes as homogeneous nodes when running GAT and GCN. Although part of the feature information will be lost, the core features of our graph construction method are preserved, named entities, keywords, titles, and their multi-interrelation.

The results obtained from the experiment are shown in Table 3. It can be seen that with the graph we constructed, a certain extent of optimization has been obtained on GCN, GAT, and HeteGCN. However, the performance of HAN is not very satisfactory because that HAN needs to specify the meta-path in advance manually. However, HAN's dual attention mechanism still has a small improvement on the Chinese datasets with the heterogeneous information graph. The improvement of our graph construction method on GCN and GAT is more obvious, especially the 0.91% improvement of GAT-HIN on 20NG.

The experimental results are shown that the multi-interrelation heterogeneous information graph for long texts is superior. Because our method considers the relative opposition between words, the semantic relationship of the article is not only be more intuitively accepted by humans and allows the model to learn more semantic information. Another benefit is that two long-distance but related words are allowed to be associated together, which makes the global semantics richer.

4.4.3 The analysis of the multi-semantic passing framework

Our heterogeneous information graph constructed for the semantic information of long texts has certain advantages in long text classification tasks. We design a multi-semantic passing framework for the constructed heterogeneous information graph to capture deeper semantics and more structural information. Moreover, we believe that the multi-semantic passing framework has a certain degree of adaptability, which can also capture more semantic information in other heterogeneous graphs.

The process of the experiment is as follows. Firstly, we select three graph construction methods for text classification. 1) Based on the most primitive graph construction method of document-document, the edges are constructed according to the similarity between documents. The proposed approach was used in GCN. 2) The document-word-based graph construction method mentioned in TextGCN. It constructs edges according to word co-occurrence and word frequency. 3) The graph construction method based on the unique words has appeared in the text and the co-occurrence mechanism of words proposed by TextING [33]. However, it is a separated graph construction method for each text. The method is adopted inside the text in this experiment, which is used the word co-occurrence mechanism to realize the interaction between texts. Secondly, we process four text data sets through these three graph construction methods to construct corresponding corpus information graphs. Thirdly, we compare their original algorithm with the multi-semantic passing framework. In particular, only text-word graphs can be considered heterogeneous among these three graph construction methods, and the other two are homogeneous.

Table 3 Test accuracy and F1 score of different methods with the HIN we constructed

Dataset		TextGCN	TextGCN-HIN	GAT	GAT-HIN	HeteGCN	HeteGCN-HIN	HAN	HAN-HIN	Han-LT
ThuCNews	*Acc*	96.82	97.12(+0.3)	97.54	97.93(+0.39)	97.95	98.11(+0.16)	97.79	97.85(+0.06)	**98.33**
	*F*1	96.6	96.83(+0.23)	97.28	97.46(+0.18)	97.69	97.76(+0.07)	97.51	97.53(+0.02)	**97.82**
SougouNews	*Acc*	97.34	97.86(+0.52)	97.84	98.33(+0.49)	98.3	98.38(+0.08)	98.06	98.11(+0.05)	**98.69**
	*F*1	97.02	97.43(+0.41)	97.33	97.75(+0.42)	98.04	98.04(-)	97.52	97.61(+0.09)	**98.21**
20NG	*Acc*	85.69	86.15(+0.46)	86.19	86.85(+0.66)	87.07	87.19(+0.12)	86.45	86.29(-0.16)	**87.62**
	*F*1	85.15	85.62(+0.47)	85.73	86.31(+0.58)	86.65	86.84(+0.19)	86.08	86.01(-0.07)	**87.14**
Ohsumed	*Acc*	68.36	68.91(+0.55)	62.56	63.47(+0.91)	65.74	66.07(+0.33)	63.2	63.03(-0.17)	**71.46**
	*F*1	67.92	68.46(+0.54)	61.94	62.73(+0.79)	65.13	63.2(+0.19)	62.67	62.53(-0.14)	**70.83**

The bold entries shows the best results of the experiments to better demonstrate the effect of Han-LT

Table 4 Test accuracy and F1 score of different graph construction methods with the multi-semantic passing framework

Dataset		GCN	GCN-M	TextGCN	TextGCN-M	TextING	TextING-M
ThuCNews	Acc	95.34	(+0.22)	96.82	(+0.25)	96.95	(+0.36)
	$F1$	94.86	(+0.23)	96.6	(+0.19)	96.52	(+0.28)
SougouNews	Acc	95.42	(+0.16)	97.34	(+0.27)	98.15	(+0.33)
	$F1$	95.01	(+0.12)	97.02	(+0.21)	97.76	(+0.29)
20NG	Acc	82.49	(+0.31)	85.69	(+0.3)	86.85	(+0.32)
	$F1$	82.04	(+0.25)	85.15	(+0.26)	86.32	(+0.27)
Ohsumed	Acc	65.73	(+0.12)	68.36	(+0.37)	64.9	(+0.44)
	$F1$	65.17	(+0.07)	67.92	(+0.24)	64.63	(+0.51)

In response to this problem, we treat the two graphs of text-text and word-word as homogeneous. Specifically, we change A_τ in the formula to A. Here, all nodes are adopted the same transformation matrix.

The final results are shown in Table 4. The multi-semantic passing framework achieves better results than the original method on these three graphs, especially in the word-word graph. We notice that in the document-document graph, the improvement effect is relatively insignificant after we apply the multi-semantic passing framework. It is because the homogeneous graph is constructed only from documents. The semantic information captured by the multi-semantic passing framework is more graph-based and global than the semantic information inside the text. The multi-semantic passing framework we designed for the long texts graph construction method is more sensitive to the capture of text semantics. Meanwhile, it can capture more structural information about the graph. Our multi-semantic passing framework can better extract semantic and structural information in both homogeneous and heterogeneous graphs.

The above three experimental results are shown that the Han-LT method has obvious superiority on long text classification tasks. Moreover, the multi-interrelation heterogeneous information graph construction method and the multi-semantic passing framework in Han-LT are flexible and applicable. The main reasons are referred Han-LT works well are twofold: 1) Multi-interrelation heterogeneous information graph based on entities, titles, and keywords can better preserve more significant semantic and structural information. 2) Based on the constructed heterogeneous information graph, the multi-semantic passing framework can adaptively find important nodes and extract more crucial higher-order semantic information to represent the long text. In conclusion, Han-LT shows sufficient superiority over other methods.

4.5 Effects of labeled data size

Nowadays, the training set of long text is difficult to construct. Therefore, there is an urgent need to develop semi-supervised long text classification methods. It is obtained that Han-LT can produce good results in semi-supervised learning due to its superior local label transfer ability. It can achieve better results than other methods in limited labeled data. We design the following experiment to verify the effectiveness of our semi-supervised long text classification method.

We chose 4 related algorithms: CNN-rand, TextGCN, HAN, and Han-LT, and the effect of the number of labeled documents are studied. Specifically, we vary the proportion of labeled texts on each dataset, and compare their accuracy on all datasets. The proportion of labeled data is increased from 2% to 30%. In addition, the experimental results are the average values obtained by running each algorithm 10 times.

From Fig. 4, it can be seen that all algorithms' accuracy on all datasets increases with the increased ratio of labeled data. Generally, the GNN-based methods achieve better accuracy, which indicates that GNN-based methods can better use limited labeled data through a message passing framework. When the proportion of labeled data is low, the performance of other algorithms drops significantly. Meanwhile, our method still maintains a relatively high accuracy, which shows that the method can better utilize limited annotated data to achieve better results in long text classification. It is because the multi-semantic passing framework can adaptively learn the importance of different nodes, which can better spread the label information of nodes locally. The superiority of Han-LT in semi-supervised learning can achieve satisfactory results even when labeled data is relatively rare.

4.6 Ablation analysis

To further verify the superiority of Han-LT on semi-supervised learning and find out which part of Han-LT provides a greater impact on semi-supervised learning, we design an ablation experiment as follows. Specifically, we divide Han-LT into two parts: the multi-interrelation heterogeneous information graph and the multi-semantic passing framework. We name the method of preserving the multi-interrelation heterogeneous information graph as

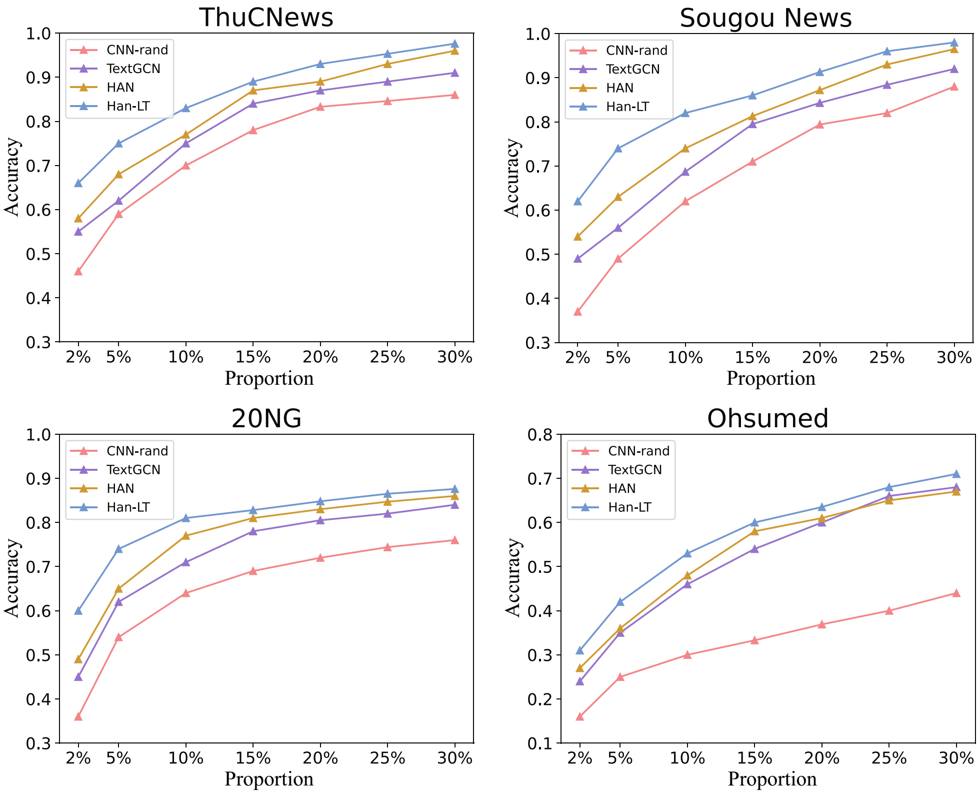


Fig. 4 The test accuracy with different proportion of labeled documents on four data sets

Han-LT_mhg, and the method of preserving the multi-semantic passing framework as Han-LT_mpf. The scale of labeled text on each dataset is varied and compared the accuracy of Han-LT, Han-LT_mhg, and Han-LT_mpf on THUCNews and 20NG. The proportion of labeled data is set as 2%, 5%, 10%, 15%, 20%, 25%, and 30%.

The experimental results are shown in Fig. 5. It is seen that the improvement of Han-LT_mhg with the multi-interrelation heterogeneous information graph is more pronounced when the proportion of labeled data is low. As the labeled data increases, the Han-LT_mpf method with the multi-semantic passing framework also becomes obvious. The improvement of Han-LT_mhg is evident because the information in the multi-interrelation heterogeneous information graph is compact. We link the core information of each article by multi-interrelation, and articles of the same category will directly become neighbors to each other with a high probability. To a certain extent, we can think of the multi-interrelation heterogeneous information graph as a ”pre-categorized” graph with core informational elements and relationships. It is because we do not make major changes to the core network structure but focuses on improving the network’s ability to perceive multi-interrelation. It is worth noting whether adding the multi-interrelation heterogeneous information graph or the multi-semantic passing framework has a particular improvement effect compared to the original GAT and TextGCN methods. In general, our proposed Han-LT method is a ensemble including the multi-interrelation heterogeneous information graph and the multi-semantic

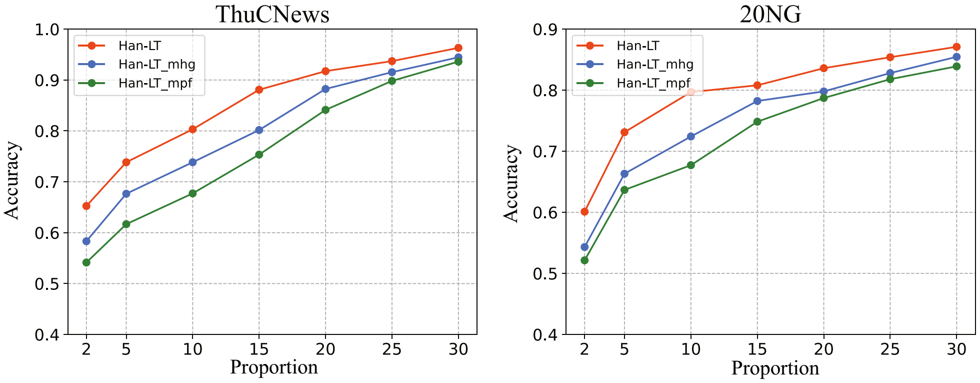


Fig. 5 Test accuracy of Han-LT variants on datasets with different proportion of labeled documents

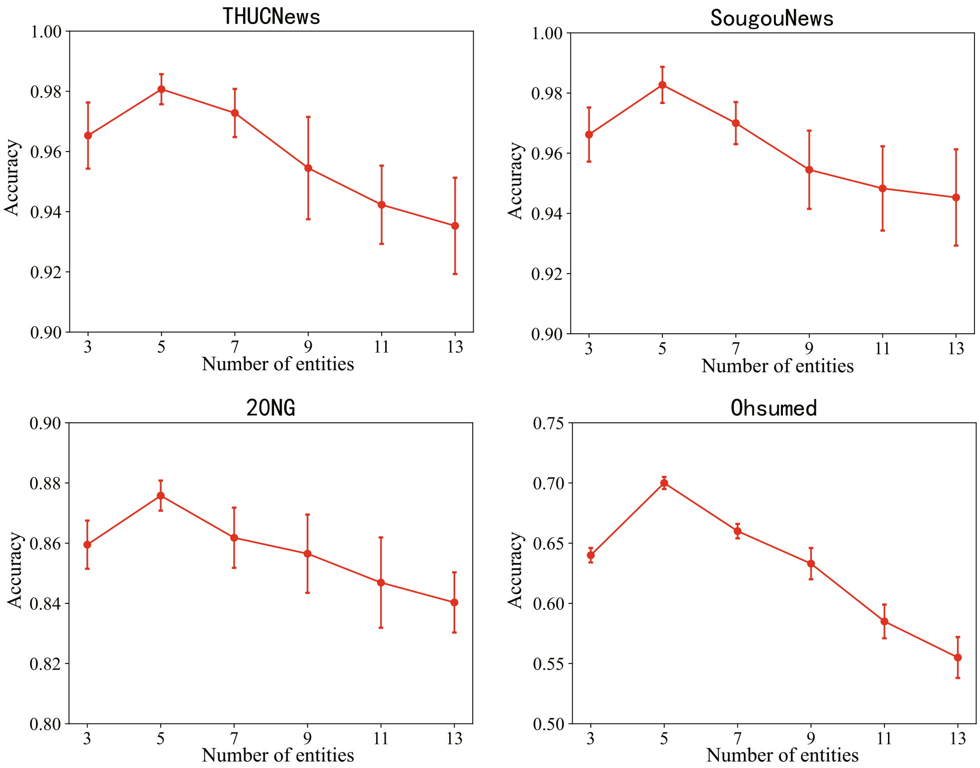


Fig. 6 The average accuracy with different number of entities on four data sets

passing framework. The multi-interrelation heterogeneous information graph connects the core elements of the article, and the multi-semantic passing framework captures the essential semantics on the graph. The combination of the two makes Han-LT provides the superiority of semi-supervised learning.

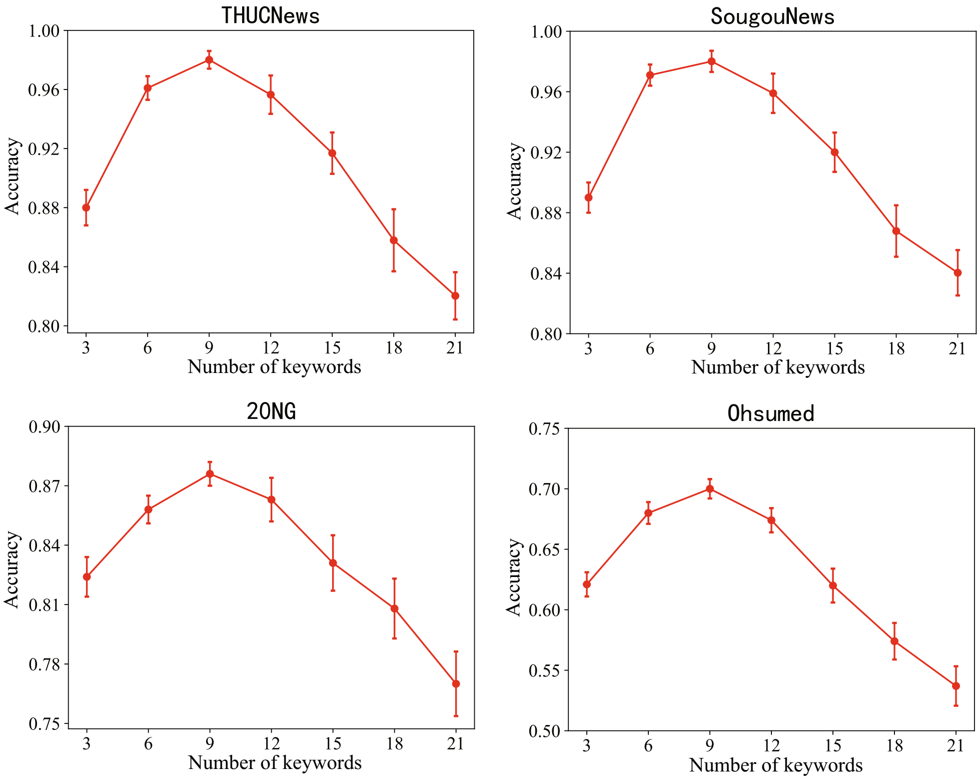


Fig. 7 The average accuracy with different number of keywords on four data sets

4.7 Parameter analysis

For Han-LT, the selection of entities and keywords is essential, which determines the difficulty of capturing semantic information and the running time of the algorithm. To intuitively show the process and ideas of our graph construction method, the following experiment is designed and visualized the results for reference. It is experimented with separately by varying the number of keywords and entities extracted for each text at each run. We set the extraction number of entities to 3, 5, 7, 9, 11, and 13 due to the high importance with low occurrence in the text. The extracted number of keywords is set to 3, 6, 9, 12, 15, 18, and 21.

Figure 6 shows the accuracy tests using different numbers of entities on the four datasets. Figure 7 shows the accuracy tests using different numbers of keywords on the four datasets. It can be seen that the accuracy rate increases with the number of selected entities and keywords in all datasets at the beginning of the experiment. However, when the number of selected entities in 20NG is greater than 5 or the number of selected keywords is greater than 10, the accuracy rate decreases with the increase of selected entities or keywords numbers. It is because selecting too many words will make the constructed heterogeneous information graph complicated. Furthermore, it is added edges between nodes that are not closely related so that the model can not accurately extract the semantics of the text. Combining the best experimental performance, we finally select 5 entity counts and 10 keyword counts.

In order to comprehensively evaluate our proposed method Han-LT, we designed the above six experiments in total. Combined with all the obtained experimental results, we strongly demonstrate that the Han-LT method can effectively preserve and extract the semantic and structural information of long texts to achieve excellent results in semi-supervised learning task. Overall, Han-LT shows significant superiority in semi-supervised long text classification tasks.

5 Conclusion

This paper proposes a semi-supervised long text classification method based on a graph neural network. Aiming at the core intention expressed by the text, we construct the long text corpus from three aspects: title, entity, and keyword. We model the text itself and link different texts together through their multi-interrelation to condense the meaning expressed by the texts while retaining the semantic structures. Then, we design the message passing framework by combining the attention mechanism and the semantic degree for the relationship between title-entity-keyword again to aggregate the multi-interrelation heterogeneous information graph. These make our model have a strong ability to extract deeper semantic and structural information. Validated by extensive experiments, our method achieves remarkable results on long text classification tasks.

Acknowledgements This work is supported by National Natural Science Foundation of China (Grant No. 61802444); the Changsha Natural Science Foundation (Grant No. kq2202294), the Research Foundation of Education Bureau of Hunan Province of China (Grant No. 20B625, No. 18B196, No. 22B0275); the Research on Local Community Structure Detection Algorithms in Complex Networks (Grant No. 2020YJ009).

Declarations

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Chen J, Gong X, Qiu Y et al (2021) Multi-label classification of long text based on key-sentences extraction. In: International conference on database systems for advanced applications, pp 3–19
2. Conneau A, Schwenk H, Barrault L et al (2016) Very deep convolutional networks for text classification. arXiv:160601781
3. Devlin J, Chang MW, Lee K et al (2018) Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 4171–4186
4. Du J, Huang Y, Moilanen K (2020) Pointing to select: a fast pointer-lstm for long text classification. In: Proceedings of the 28th international conference on computational linguistics, pp 6184–6193
5. Du J, Huang Y, Moilanen K (2021a) Knowledge-aware leap-lstm: integrating prior knowledge into leap-lstm towards faster long text classification. In: Proceedings of the AAAI conference on artificial intelligence, pp 12768–12775
6. Du J, Vong CM, Chen CLP (2021b) Novel efficient rnn and lstm-like architectures: recurrent and gated broad learning systems and their applications for text classification. IEEE Trans Cybern 51(3):1586–1597
7. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Advances in Neural Information Processing Systems 30
8. Johnson R, Zhang T (2017) Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th annual meeting of the association for computational linguistics, pp 562–570
9. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp 1746–1751
10. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv:160902907
11. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems
12. Lan Z, Chen M, Goodman S et al (2020) Albert: a lite bert for self-supervised learning of language representations. In: International conference on learning representations

13. Linmei H, Yang T, Shi C et al (2019) Heterogeneous graph attention networks for semi-supervised short text classification. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, pp 4821–4830
14. Liu P, Qiu X, Huang X (2016) Recurrent neural network for text classification with multi-task learning. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, pp 2873–2879
15. Mikolov T, Chen K, Corrado G et al (2013) Efficient estimation of word representations in vector space. Proceedings of workshop at ICLR
16. Minaee S, Kalchbrenner N, Cambria E et al (2021) Deep learning–based text classification: a comprehensive review. ACM Comput Surv 54(3):1–40
17. Monti F, Otness K, Bronstein MM (2018) Motifnet: a motif-based graph convolutional network for directed graphs. In: 2018 IEEE data science workshop, pp 225–228
18. Peng H, Li J, Wang S et al (2021) Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification. IEEE Trans Knowl Data Eng 33(6):2505–2519
19. Radford A, Narasimhan K (2018) Improving language understanding by generative pre-training
20. Ragesh R, Sellamanickam S, Iyer A et al (2021) Hetegcn: heterogeneous graph convolutional networks for text classification. In: Proceedings of the 14th ACM international conference on web search and data mining, pp 860–868
21. Tan Z, Chen J, Kang Q et al (2022) Dynamic embedding projection-gated convolutional neural networks for text classification. IEEE Trans Neural Netw Learn Syst 33(3):973–982
22. Tang J, Qu M, Mei Q (2015) Pte: Predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1165–1174
23. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. Advances in Neural Information Processing Systems 30
24. Veličković P, Cucurull G, Casanova A et al (2018) Graph attention networks. In: International conference on learning representations
25. Wang X, Ji H, Shi C et al (2019) Heterogeneous graph attention network. In: The world wide web conference, pp 2022–2032
26. Weijie D, Yunyi L, Jing Z et al (2021) Long text classification based on bert. In: 2021 IEEE 5th information technology, networking, electronic and automation control conference (ITNEC), pp 1147–1151
27. Yang T, Hu L, Shi C et al (2021) Hgat: heterogeneous graph attention networks for semi-supervised short text classification. ACM Trans Inf Syst 39(3):1–29
28. Yang Z, Yang D, Dyer C et al (2016) Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1480–1489
29. Yang Z, Dai Z, Yang Y et al (2019) Xlnet: generalized autoregressive pretraining for language understanding. Advances in Neura l Information Processing Systems 32
30. Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence, pp 7370–7377
31. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. arXiv:14092329
32. Zhang C, Song D, Huang C et al (2019) Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 793–803
33. Zhang Y, Yu X, Cui Z et al (2020) Every document owns its structure: inductive text classification via graph neural networks. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 334–339

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

Wei Ai received the Ph.D. degree in the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. Her research interests include date mining, big data, cloud computing, and parallel computing.

Ze Wang is currently studying for a master's degree. School of computer and information engineering, Central South University of forestry and technology, Changsha, China. His research interests include natural language processing and text classification.

Hongen Shao is currently studying for a master's degree. School of computer and information engineering, Central South University of forestry and technology, Changsha, China. His research interests include natural language processing and named entity recognition.

Tao Meng received the Ph.D. degree in the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include date mining, Network analysis.

Keqin Li is a SUNY Distinguished Professor of Computer Science with the State University of New York. He is also a National Distinguished Professor with Hunan University, China. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyberphysical systems, heterogeneous computing systems, big data computing, highperformance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has authored or coauthored over 880 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top 5 most inflfluential scientists in parallel and distributed computing in terms of both single-year impact and career-long impact based on a composite indicator of Scopus citation database. He has chaired many international conferences. He is currently an associate editor of the ACM Computing Surveys and the CCF Transactions on High Performance Computing. He has served on the editorial boards of the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Services Computing, and the IEEE Transactions on Sustainable Computing. He is an IEEE Fellow and an AAIA Fellow. He is also a Member of Academia Europaea (Academician of the Academy of Europe).

Affiliations

Wei Ai[1] · **Ze Wang**[1] · **Hongen Shao**[1] · **Tao Meng**[1] · **Keqin Li**[2]

Wei Ai
aiwei@hnu.edu.cn

Ze Wang
zewang@csuft.edu.cn

Hongen Shao
hongen.shao@csuft.edu.cn

Keqin Li
lik@newpaltz.edu

[1] School of Computer and Information Engineering,
Central South University of Forestry and Technology,
Changsha, 410082, Hunan, China

[2] Department of Computer Science, State University
of New York, New Paltz, NY 12561, USA