International Journal for Housing Science and Its Applications
Publish August 5, 2025. Volume 46, Issue 3 Pages 4586-4598

https://doi.org/10.70517/ijhsa463393

Modeling Analysis of Modern Chinese Quantitative Word Constructions Based on Graph Neural Networks

Jianping Xu[1,*]
[1] Tianjin University of Finance and Economics Pearl River College, Tianjin, 300000, China
Corresponding authors: (e-mail: jpxrunning@163.com).

Abstract In the transformation process of traditional media to converged media, the labeling technology of modern Chinese language becomes more important, and the labeling of modern Chinese text is not only convenient for organizing and categorizing, but also can provide more accurate search and recommendation services for Internet users. In this paper, a global classification model for multi-labeled text is constructed by combining graph convolutional neural network, multi-head attention mechanism and BERT pre-training model, so as to realize the modeling of quantitative word constructions in modern Chinese. In the experimental datasets, the classification accuracies of the models after BiGRU and BERT are added to GCN are significantly improved, while the classification accuracy of the BERT+GCN model in this paper is better than that of the BiGRU+GCN model, which verifies the effectiveness of the text classification model in this paper. In addition, the classification effect of this paper's method on four datasets is better than all other compared models, and it improves 1.31%, 0.98%, 1.44%, and 0.50% compared to BiHAM model on the four datasets of Ohsumed, MR, R52, and R8, respectively. The model application results show that both "length" and "length and short" can be collocated with quantifiers, there are some common collocation quantifiers between the two, and the collocation of the two quantifiers also exists two positions, but the former is more significant in this feature. This paper provides an analytical path for modeling quantity word constructions in modern Chinese.

Index Terms graph convolutional neural network, BERT, attention mechanism, text categorization, quantifier constructions, modern Chinese language

I. Introduction

Modern Chinese is a language spoken by the modern Han Chinese nation, including a variety of dialects and national common languages [1], [2]. In Modern Chinese, quantifier constructions have always been the focus of scholars' research [3]. Conjugation is a simplified form of Chinese lexical and syntactic structure, which is influenced by both phonological and semantic structure [4], [5]. The use of constructs not only improves the efficiency of information communication, but also enhances the accuracy of information [6]. Constructions play an important role in rapid verbal communication, concise textual expression and precise semantic communication [7], [8]. Although its use has been influenced by various levels of pragmatics and linguistics, it has always been an important form of Chinese language expression and occupies an important position in modern people's communication [9]-[11].

In addition, the use of constructions can make new generalizations and explanations of some special sentences in modern Chinese [12]. Such sentences are exemplified or elaborated by double quantifier constructions in the form of noun phrase with quantifier + verb + noun phrase with quantifier, with the meaning of the construction of quantity supply and demand equilibrium, i.e., the quantity supplied is greater than or equal to the quantity demanded, and the supply and demand are balanced, and the demand is satisfied [13]-[16]. Constructs override the original meaning of the verb, forbidding verbs that are contrary to the meaning of the construct, with some tolerance and compromise for other verbs, and the formation and use of this construct can be explained by the use of speech use theory.

In this paper, we propose a multi-label text classification model for modern Chinese that incorporates graph convolutional networks, which is used to solve the problem of insufficient feature fusion between labels and text. Aiming at the tree structure of labels, the model uses graph convolutional neural networks to model the dependencies between labels, and the extracted label features are incorporated into the text features using a multi-attention mechanism to take full advantage of the interactions between the text features and the label features, in order to improve the performance of the model on the modern Chinese dataset. In order to verify the performance of the model, it is evaluated and applied to the analysis of quantifier constructions of synonymous attribute nouns “length” and “long and short”.

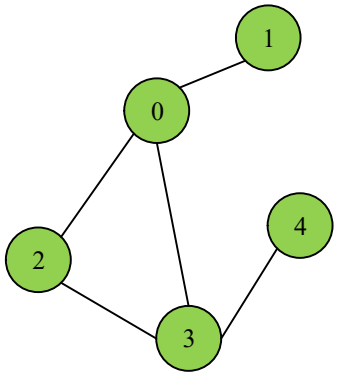
II. Global Classification Model for Multi-Labeled Text by Fusing Graph Neural Networks

In order to model and analyze modern Chinese quantitative word constructions, this paper proposes a global classification model for multi-labeled text by fusing graph neural networks.

II. A. Graph Neural Networks

Graph-structured data is a class of data form widely existed in real life, and many data are suitable to be recorded in the form of graphs, such as the social circle of users in the network, routing in communication, transportation and so on. The graph neural network is able to better utilize the graph data and input the data in the form of graph into the neural network for learning, which can enrich the features of the input data and extend the breadth of the application.

The input needed for GNN [17] is the graph $G=(V,E)$, where graph specifically refers to a form containing nodes V and edges E. The graph in a graph neural network is shown in Figure 1. In Fig. 1(a), the circles represent the nodes and the lines connecting them represent the edges, the graph is an undirected cyclic graph containing five nodes and five edges, in GNNs it is common to utilize adjacency matrices, degree matrices, and Laplace matrices, etc., to store information contained in the graph, and the adjacency matrix corresponding to Fig. 1(b) can be represented as Fig. 1(a). There are five nodes in Fig. 1(a), which are nodes 0, 1, 2, 3, and 4, and node 0 is connected to nodes 1, 2, and 3 as undirected edges, which are represented in the adjacency matrix as the six coordinates (0, 1), (0, 2), (0, 3), (1, 0), (2, 0), and (3, 0), and the connections of edges are entered into the model and computed as adjacency matrices. Each node and edge can be given certain initial features when inputting into the GNN, in natural language processing, node features are usually pre-trained text embeddings, while edge features are usually relationships between nodes, and in Fig. 1 edges are given a weight of 1.



(a) Graph

0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
0	0	0	1	0

(b) Adjacency matrix

Figure 1: Graph in graph neural networks

By learning node embeddings as well as graph structures, GNN can be applied to node-level tasks as well as graph-level tasks, and the learning process can be summarized as shown in equation (1):

$$h_t^{(l)} = f_{filter}\left(A, H^{(l-1)}\right) \tag{1}$$

where A denotes the adjacency matrix, $H^{(l-1)}$ denotes the state of the node in the previous layer, and $H^{(l-1)}$ denotes the initial feature of the node when $l=1$, which is the node embedding. The f_{filter} denotes the graph filter, i.e., the way the node updates its state, and different choices of f_{filter} determine different GNN models.

GraphSAGE employs two steps of sampling and aggregation on the graph structure to learn the node information. First a central node is identified, the neighboring nodes are sampled by relying on the adjacency matrix,then the information of the neighboring nodes and the central node is aggregated by a multilayer aggregation function,and finally the label of the node is predicted with the aggregated information.The specific working of the GraphSAGE model is shown in Fig. 2,which adopts a two-layer aggregation method.

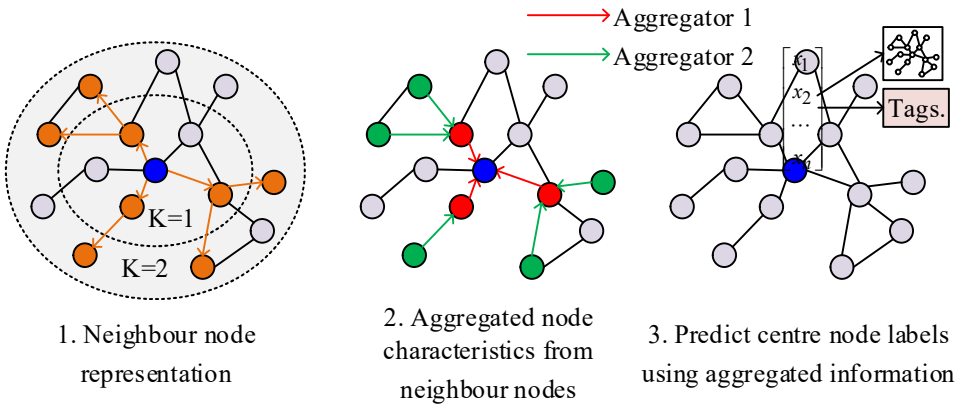


Figure 2: GraphSAGE model

Four node aggregation functions are provided in the GraphSAGE model:

(1) Mean value aggregation: when aggregating the central node, the mean value of the feature vectors of the neighboring nodes is taken, see equation (2):

$$h_v^k \leftarrow Mean\left(\left\{h_u^{k-1}, \forall u \in N(v)\right\}\right) \tag{2}$$

where h_v^k represents the features of v node at k th layer, and $N(v)$ represents the set of neighbor nodes of v node.

(2) Graph Convolutional Aggregation: when aggregating the center nodes, the convolution values of the feature vectors of the neighboring nodes and the center node are taken and then averaged, and finally the obtained results are nonlinearly transformed, see equation (3):

$$h_v^k \leftarrow \sigma\left(W \cdot Mean\left(\left\{h_v^{k-1}\right\} \cup \left\{h_u^{k-1}, \forall u \in N(v)\right\}\right)\right) \tag{3}$$

(3) Long-short-time aggregation: since GraphSAGE constructs an undirected graph, the order of the nodes in the graph is disrupted and put into the LSTM model.

(4) Pooling aggregation: when aggregating the central nodes, the features of the neighboring nodes are first put into a fully connected layer and then aggregated using the maximum pooling function, see equation (4):

$$h_v^k \leftarrow \max\left(\left\{\sigma\left(W_{pool} h_u^{k-1} + b\right), \forall u \in N(v)\right\}\right) \tag{4}$$

The GraphSAGE model proposes both supervised learning as well as unsupervised learning, where supervised learning calculates the loss value using a function that can satisfy the prediction objective, such as the cross-entropy loss function. Unsupervised learning, on the other hand, proposes an assumption that the central node and its neighboring node u have similar feature vectors, while the feature vectors of several points that are not similar to its neighbors are not similar, and the proposed loss function is shown in Equation (5):

$$J_G(z_u) = -\log\left(\sigma\left(z_u^T z_v\right)\right) - Q \cdot E_{v_n \sim p_n(v)} \log\left(\sigma\left(-z_u^T z_{v_n}\right)\right) \tag{5}$$

where z_u is the feature vector after GraphSAGE aggregation, while v represents the neighboring nodes obtained by node u after random wandering, $v_n \sim P_n(v)$ denotes negative sampling, and Q denotes the number of samples.

II. B.Pre-training model

II. B. 1) Attention mechanisms

The core principle of the attention mechanism is to dynamically focus on key information by calculating the relevance weights of each part of the input sequence to the task at hand. This process can be formalized as a Query, Key and Value matching game. Specifically, for a given input sequence, the corresponding Query (Q), Key (K) and Value (V) vectors are obtained by linear transformation. These vectors are obtained by multiplying the input sequences with the corresponding weight matrices, and the similarity between the query vectors and all the key vectors is computed

as a basis for the attention score. This step is usually implemented by dot product or other similarity computation methods and scaled to prevent the problem of vanishing gradients. Finally, a softmax function is applied to convert these scores into probability distributions, ensuring that the attention weights for each part sum to 1.

Finally, the value vectors are weighted and summed according to these probability distributions to obtain a weighted attention representation. This representation captures the most critical information in the input sequence and focuses this information on the current processing stage of the model. In a multi-head attention mechanism, the above process is executed several times in parallel, with each head capturing a different aspect of the input sequence, and then the outputs of all heads are combined to form a comprehensive attention representation.

The proposal and application of the attention mechanism greatly improves the ability of NLP models to process long sequential data, especially in capturing long-distance dependencies. It enables the model to not only process local information but also make decisions in a global context.

II. B. 2) Transformer model

The Transformer model [18] abandons the traditional recurrent neural network structure and adopts a self-attentive mechanism to process sequence data, which enables the model to consider information from all positions in the sequence simultaneously, thus effectively capturing long-range dependencies.

In the architecture of Transformer, the whole model consists of two parts: encoder and decoder. The encoder consists of multiple identical layers stacked on top of each other, each containing a self-attention sublayer and a feed-forward neural network sublayer. The self-attention sublayer computes the attention score using three matrices, Query, Key and Value, which are obtained by multiplying the input matrix with the learnable weight matrix. The formula for calculating the attention score is given below:

$$Attention\left(Q,K,V\right)=soft\max\left(\frac{QK^{T}}{\sqrt{d_{k}}}\right)V \tag{6}$$

The softmax function converts the scores into a probability distribution, ensuring that each column sums to 1.

Transformer further employs the multi-head self-attention mechanism, which enhances the model's ability to capture information from different locations by computing the attention of multiple heads in parallel. The output of the multi-head self-attention is the result of stitching together the outputs of all the heads as shown in Equation (7):

$$MultiHead\left(Q,K,V\right)=Concat\left(head_{1},\ldots,head_{h}\right)W^{O} \tag{7}$$

Here, $MultiHead$ denotes the attentional output of a single head, W^{O} is the learnable weight matrix, h is the number of heads, and $Concat$ is the splicing operation.

Each layer of the encoder and decoder employs residual join and layer normalization techniques. Residual concatenation helps mitigate the problem of vanishing gradients in deep network training by skipping certain layers and adding inputs directly to the outputs of the layers. Layer normalization, on the other hand, improves model stability and training speed by normalizing the output of each layer.

Since the Transformer model does not contain a loop structure, it cannot naturally process the position information in the sequence as RNN does. To solve this problem, Transformer introduces position coding, which is implemented by adding position information to each element of the input sequence. Position encoding is typically generated using a combination of sine and cosine functions, which maintains the order information of the elements in the sequence.

II. B. 3) BERT model

The core of the BERT model [19] lies in the use of large amounts of unlabeled text for pre-training, which leads to the learning of deep linguistic representations that can subsequently be adapted to a variety of specific downstream tasks through a process of fine-tuning.

The input to the BERT model consists of three components: word embeddings, positional embeddings and segmental embeddings. The word embeddings map each word in the glossary to a fixed-dimension vector space, while the position embeddings provide the model with information about the position of the words in the sequence. Segment embeddings are used to distinguish between different sentences in the input sequence, which is introduced when dealing with e.g., question and answer tasks.

The network structure of BERT is based on the encoder part of the Transformer, which consists of multiple layers of Transformer blocks. Each Transformer block contains two main sub-layers: a self-attention layer and a feed-forward neural network layer. The self-attention layer uses a multi-head attention mechanism to process all words in a sequence in parallel, allowing the model to consider information from all positions simultaneously. The output of multi-head attention is obtained by splicing and linearly transforming the output of each head.

Pre-training for BERT consists of two tasks: masked language modeling and next sentence prediction (NSP). In masked language modeling, the goal of the model is to predict randomly masked words. Specifically, the MLM task is performed as follows: first, sentences are randomly selected from a large amount of text data, and then some words are randomly selected in each sentence and replaced with a special [MASK] token. This [MASK] marker is a placeholder indicating that the word at that position has been masked, requiring the model to predict what the original word was. During training, the BERT model receives such sentences with [MASK] markers and tries to predict the original word at each [MASK] position.

The NSP task is one of the key components of the BERT pre-training process, and this task is designed to allow the model to learn sentence-level representations by better understanding the relationships between sentences. In the NSP task, the model receives a pair of sentences as input and needs to predict whether the second sentence is the next sentence immediately following the first.

The pre-training process of BERT is performed on a large amount of text, allowing the model to learn a rich linguistic representation. After the pre-training is completed, BERT can be adapted to specific downstream tasks through a fine-tuning process. During the fine-tuning process, BERT is trained on task-specific datasets to optimize the task-related loss function.

II. C. Global classification model incorporating graph neural networks

In this paper, we make the global algorithm, i.e., in hierarchical multi-label classification problems, only one classifier is trained and only one loss function is optimized by considering the overall structure of the labels.

II. C. 1) Network structure

The overall structure of the global classification model is shown in Fig. 3. Studies have shown that the text representation extracted by BERT is better than that extracted using a combination of word vectors and LSTM, so in this paper, BERT is used as an encoder for the text, and the parameters are fine-tuned using pre-trained parameters in order to obtain the parameters that perform optimally on the text of modern Chinese. After the model makes embedding for each label in the label space, it is encoded using GCN [20], each node will aggregate the information of the neighboring nodes, and after the multi-layer GCN encoding, the links between the labels will be sufficiently injected into the model to obtain the label representation. In order to fuse label features with text features, the model uses a multi-head attention mechanism to inject the learned label dependencies into the text representation. The final prediction layer uses the average pooled information from the textual representation as input, which is transformed into the space of the output through full connectivity. The whole model is an end-to-end global model, which is divided into four parts, namely text representation layer, label embedding layer, feature fusion layer and prediction layer.

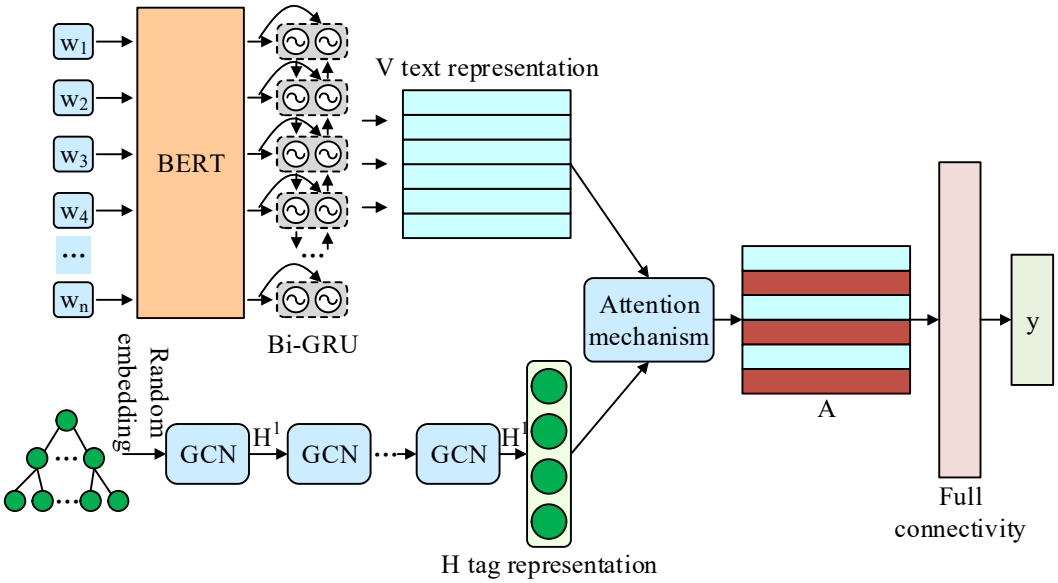


Figure 3: Structure of the global classification model

II. C. 2) Obtaining Text Representations Based on Pre-trained Models

In this paper, BERT is used for embedded representation of text without using word vectors.The input to the BERT model is shown in equation (8):

$$E = E_{word} + E_{position} + E_{segment} \tag{8}$$

where $E, E_{word}, E_{position}, E_{segment} \in \mathbb{R}^{n\times 768}$, and n is the text length.

In this chapter, $BERT_{BASE}$ was used for the experiments, which was modeled using a 12-layer Transformer composition, with a hidden unit size of 768, and a multi-head self-attentive attention with a head count of 12, for a total of 110M parametric quantities. After embedding the words in the text through the input layer and then encoding them using BERT as shown in equation (9), the output of the model is $D \in \mathbb{R}^{n\times 768}$, n is the length of the text:

$$D = BERT - Encoder(E) \tag{9}$$

When conducting experiments, the pre-trained parameters are loaded, but the effect of using such a model directly is not optimal, and the model needs to be fine-tuned according to the actual situation of the task, so the BERT is used as the bottom layer, and the downstream task is fine-tuned by connecting it afterward, and the hyper-parameters are kept in the same line with those of the pre-training when fine-tuning.

In order to enable the model to extract the features of the text more deeply, $D = [d_1, d_2, \ldots, d_n]$ is regarded as the word embedding matrix extracted by the BERT, and d_i denotes the feature of the i th word, which is fed into the Bi-GRU to perform another feature extraction, as shown in equation (10):

$$\begin{aligned} \overrightarrow{h_i} &= \overrightarrow{GRU}\left(\overrightarrow{h_{i-1}}, d_i\right) \\ \overleftarrow{h_i} &= \overleftarrow{GRU}\left(\overleftarrow{h_{i+1}}, d_i\right) \\ h_i &= \left[\overrightarrow{h_i}, \overleftarrow{h_i}\right] \end{aligned} \tag{10}$$

where $h_i \in \mathbb{R}^{2u}$, is the result of splicing the bidirectional expression of each word in the bidirectional GRU. The encoded information of a modern Chinese text is $V = [h_1, h_2, \ldots, h_n] \in \mathbb{R}^{n\times 2u}$, and the reason for choosing GRU instead of LSTM is that the GRU model is simpler to compute.

II. C. 3) Figure Embedding Layer

In the labeling tree of modern Chinese text, each edge represents the superior-subordinate relationship between the nodes it connects, and the embedding matrix of labels can be obtained by capturing the labeling relationship using GCN.

First use the label embedding for random initialization using Kaiming uniform distribution to get $H^0 \in \mathbb{R}^{K\times d_h}$, d_h is the dimension of the label embedding, and K is the number of label categories. When the labeled tree is converted into a graph, its adjacency matrix can be expressed as A, and the adjacency matrix is summed with the unit matrix to obtain $\tilde{A}$. The feature extraction of the labeled graph is performed using GCN network, and the GCN computation at each layer is shown in equation (11), and the output after multiple layers of GCN is notated as $H^l \in \mathbb{R}^{K\times d_t}$, and d_t is the dimension of the final labeled feature:

$$H^{i+1} = \mathrm{Re}\,LU\left(\tilde{L}_{sym} \cdot H^i \cdot W_i\right) \tag{11}$$

where $\tilde{L}_{sym}$ denotes the Laplace matrix in renormalized form, $\tilde{L}_{sym} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}$, $\tilde{A} = A + I$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and W_i is a parameterized weight matrix that serves to perform an affine variation of the input graph signal matrix to enhance the fitting ability of the network.

From a space-based perspective, $\tilde{L}_{sym}H^i$ essentially performs an aggregation operation on the feature vectors of the neighboring nodes, and thus the GCN is able to learn label-to-label connections in label embedding, which encompasses upper-layer-to-lower-layer connections as well as mutual constraints between labels.

II. C. 4) Feature Fusion Layer

The feature fusion layer selects the multi-head attention mechanism to perform feature fusion between the features of the labels extracted from the graph embedding layer and the text representation extracted using BERT. The multi-head attention mechanism is the most critical feature extractor in Transformer, which transforms the query, key, and value in the attention mechanism into d_k, d_k, d_v dimensions linearly, and then transforms them multiple times to form different parallel "heads". The attention function is then used in parallel, and the results of the multiple heads are eventually merged and projected to form the final result. This mechanism allows the model to focus on information from different representation subspaces.

In this chapter, the obtained label vector H is used as the query, and the text feature matrix V extracted by BERT is used as the key and value, and the calculation of the multi-head attention is shown in equation (12):

$$\begin{aligned} W_{att} &= soft\max\left(\frac{HW_i^Q \cdot (VW_i^K)^T}{\sqrt{d_k}}\right) \\ head_i &= W_{att} \cdot VW_i^V \\ A &= \left[head_1 \oplus \cdots \oplus head_h\right] \cdot W^O \end{aligned} \tag{12}$$

where $\oplus$ denotes the splicing operation, $W_i^Q \in \mathbb{R}^{d_t \times d_k}$, $W_i^K \in \mathbb{R}^{2u \times d_k}$, and $W_i^V \in \mathbb{R}^{2u \times d_v}$ for the parameters in each parallel attention, $W^O \in \mathbb{R}^{hd_v \times 2u}$ for the parameters of the final linear transformation, and the number of heads of the parallel attention used is h, $d_v = d_k = \frac{2u}{h}$. The resulting A is the matrix that incorporates the label embedding and text features.

II. C. 5) Forecasting layer

In multi-label classification, the last layer usually uses a sigmoid function rather than a softmax function, which in essence utilizes the idea of problem transformation to transform a multi-label problem into multiple binary classification problems, and thus expresses in each dimension of the output whether or not the sample belongs to the corresponding class, whose value can be regarded as the probability of belonging to the corresponding class. The task of the prediction layer is to make A transform into such a probability vector.

To perform classification, the prediction layer maps the output to the label space using two layers of full connectivity, outputting a global prediction as shown in equation (13):

$$\begin{aligned} A_g &= \mathrm{Re}LU\left(W_f \cdot avg(A)^T + b_f\right) \\ \hat{y} &= \sigma\left(W_g A_g^T + b_g\right) \end{aligned} \tag{13}$$

where $W_f \in \mathbb{R}^{k \times 2u}$, $b_f \in \mathbb{R}^{k \times 1}$ are the parameter and bias vectors of the first fully connected, k is the number of neurons in the fully connected layer, respectively, $W_g \in \mathbb{R}^{K \times k}$, $b_g \in \mathbb{R}^{K \times 1}$ is the parameter and bias vector of the last layer, and $\sigma(\cdot)$ is the sigmoid activation function.

The computed $\hat{y}$ is a continuous vector, and the value $\hat{y}^i$ at each position in this vector denotes the probability that a sample of modern Chinese belongs to the class l_i, $P(l_i \mid x, \gamma)$, $l_i \in \mathrm{L}$.

II. C. 6) Loss function

During training, a binary cross-entropy loss function (BCE) is used for optimization, and the loss function is shown in equation (14):

$$\mathrm{L} = -\frac{1}{M}\sum_{i=1}^{M} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i) \tag{14}$$

where y is the true label and $\hat{y}$ is the output predicted by the model.

In order to take into account the hierarchical dependence of the labels in the global output, this chapter uses a recursive regularization approach. The method adds a regularization term to the final loss function so that similar categories in the hierarchy have similar parameters. The regularization method is shown in equation (15):

$$\lambda(\Phi)=\sum_{t\in \mathrm{L}}\frac{1}{2}\left\|w_t-w_{\pi(t)}\right\|^2 \tag{15}$$

where $\Phi=\{w_l : l\in \mathrm{L}\}$ is the parameter of the fully-connected layer of the final prediction function, with the parameter w_l corresponding to each label l, and $\pi(l)$ denotes the node l is the mother node in the label tree.

The above two components are summed to obtain the final cost function as shown in equation (16):

$$\mathrm{J}\ (\Theta)=\mathrm{L}+c*\lambda(\Phi) \tag{16}$$

where c is the hyperparameter of the penalty term. During training, the model is optimized using the Adam optimizer.

III. Model performance evaluation and practical application

In this chapter, based on the performance evaluation of the proposed global classification model for text, the model is applied to model the quantifier constructions in modern Chinese, and the quantifier collocations of two nouns with synonymous attributes, namely, “length” and “long and short”, are investigated.

III. A. Model performance evaluation experiments

III. A. 1) Data sets

In this paper, Ohsumed, R52, R8, and MR are selected as the corpus datasets for the model text categorization performance evaluation experiments. Among them, the Ohsumed corpus is from the MEDLINE database, which is constructed based on 14,038 of the 20,000 document abstracts included chronologically since 1991 under the category of cardiovascular diseases in the MEDLINE literature database, which contains abstracts of cardiovascular disease-related documents covering a total of 23 disease categories. Based on the classification requirements, the documents were screened and a total of 7524 document abstracts were retained, of which 3413 abstracts constituted the training set and the remaining 4111 abstracts constituted the test set.Both R52 and R8 datasets were derived from the Reuters21578 dataset.R8 has 8 classification categories and contains 5546 documents in the training set and 2212 documents in the test set. R52 has 52 classification categories, the training set contains 6604 documents and the test set contains 2605 documents.The MR dataset is a movie review dataset for binary sentiment classification. Each review contains only one sentence, and the whole dataset contains 5442 positive and negative examples each.The specific information of Ohsumed, R52, R8, and MR datasets is shown in Table 1.

Table 1: Specific information of the Ohsumed, MR, R52, and R8 datasets

Data set	Number of texts	The number of training set documents	The number of test set documents	Vocabulary count	Number of categories	Average length of the text
Ohsumed	7524	3413	4111	15279	23	136.94
MR	10884	5442	5442	30545	2	21.47
R52	9209	6604	2605	18014	52	70.91
R8	7758	5546	2212	15473	8	66.83

III. A. 2) Experimental content

The experimental environment used in this section is conducted on Windows 11 operating system and the graphics card is an RTX3090 GPU.In the session of preprocessing the selected dataset the word feature vector dimensions are set to 300 and the training and validation sets are divided according to the ratio of 8:2.The final size of the training set obtained is 80% of the original training set, and the size of the validation set obtained is 20% of the original training set. 20%. The number of training iterations of this model is set to be 3000.In the session of training the classification model using the samples of the training set, the learning rate is set to be 0.00005, the maximum number of training iterations is 500, and the minimum number of training iterations is 10.The output dimension of the first layer of graph convolutional neural network is 250, and that of the second layer of graph convolutional neural network is 120.The Dropout rate is 0.5.

III. A. 3) Experimental results and analysis

In order to verify the effectiveness of the proposed hierarchical multi-label text categorization model in various aspects, this paper conducts ablation experiments and text categorization comparison experiments.

(1) Ablation Experiment

In order to verify the role of graph convolutional network in learning, this paper separately conducts ablation experiments on the semantic feature extraction model using word vectors plus BiGRU and the BERT model used

in this paper with or without the addition of GCN, and both of them use the mechanism of multi-head attention to incorporate the labeled representations extracted by GCN. A comparison of the text classification results of the model ablation experiments is shown in Table 2.

It can be seen that on the four experimental datasets of Ohsumed, MR, R52, and R8, the classification accuracy of the model after the addition of GCN by BiGRU has been improved by 1.39%, 1.66%, 1.65%, 1.56%, and that of the model after the addition of GCN by BERT has been improved by 1.72%, 1.22%, 1.87%, and 2.29%, respectively, and the simultaneous The classification accuracy of the BERT+GCN model in this paper is improved compared to both BiGRU+GCN model, thus verifying the effectiveness of using BERT model combined with graph convolutional neural network for text classification in this paper.

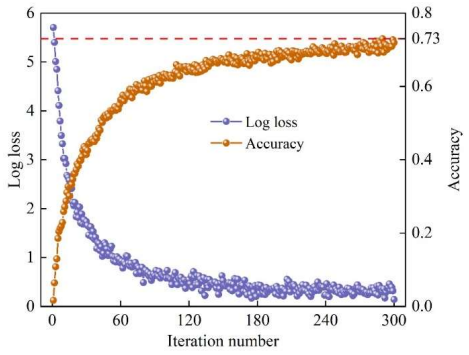
Table 2: Comparison of text classification results in model ablation experiments

Model	Classification accuracy rate /%			
	Ohsumed data set	MR data set	R52 data set	R8 data set
BiGRU	70.45	76.69	93.76	95.42
BiGRU+GCN	71.84	78.35	95.41	96.98
BERT	70.87	78.04	94.28	96.14
BERT+GCN	72.59	79.26	96.15	98.43

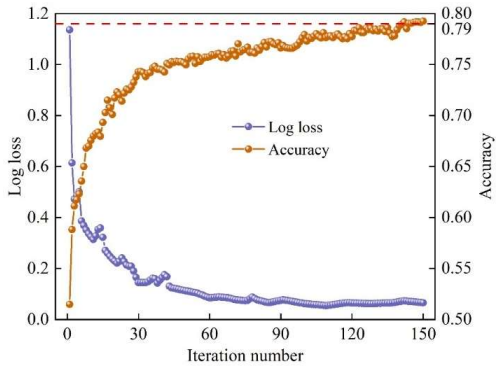
(2) Comparison experiments of text categorization

Experiments are conducted on four datasets, Ohsumed, MR, R52, and R8, and the loss rate and classification accuracy of this paper's model are shown in Fig. 4, where (a) to (d) denote the training results on the Ohsumed, MR, R52, and R8 datasets, respectively.

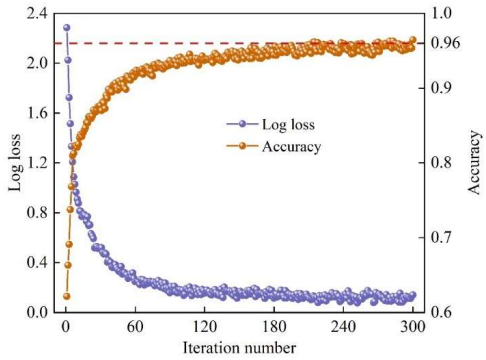
From the training results, it can be seen that the model in this paper achieves good convergence results on all four datasets. Although, on the MR, R8, and R52 datasets, the classification accuracy of the model on the corresponding datasets fluctuates in different magnitudes as the number of training iterations increases. However, the overall trends all show an increase with the gradual decrease of the loss rate and reach the best classification results at the end of training.



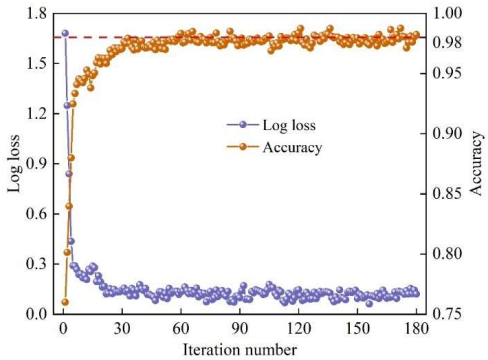
(a) Ohsumed



(b) MR



(c) R52



(d) R8

Figure 4: The loss rate and classification accuracy of the model in this paper

In order to further verify the effectiveness of the LGGCN model, the text classification models based on traditional convolutional neural networks and recurrent neural networks in recent years are selected to compare the experimental results with this paper's method on the four datasets of Ohsumed, MR, R52, and R8, and the results of the comparison of the classification accuracy of the text classification models are shown in Table 3.

It can be seen that the classification effect of this paper's method on the four datasets is better than most of the text classification models in recent years, reaching 72.59%, 79.26%, 96.15%, 98.43% on the four datasets of Ohsumed, MR, R52, and R8, respectively, which is significantly better than all the other comparative models, and compared with the BiHAM model, the classification accuracy has been improved by 1.31%, 0.98%, 1.44%, and 0.50%, respectively. The comparison results show that after using the text classification model combining BERT and graph convolutional neural network in this paper, compared with BiHAM without using the pre-trained model, it has a more powerful feature extraction ability in constructing graph relations, and can more accurately describe the content of each text on the basis of inscribing inter-text relations, and then improve the classification accuracy of text classification, which is suitable for the modern Chinese hierarchical multi labeled It is suitable for modern Chinese hierarchical multi-label text categorization task.

Table 3: Comparison of the accuracy rates of current text classification models

Model	Classification accuracy rate /%			
	Ohsumed data set	MR data set	R52 data set	R8 data set
CNN-rand	46.60	76.21	86.32	94.62
CNN-non-static	61.22	78.19	88.67	96.62
LSTM	44.29	77.16	86.47	94.65
Bi-LSTM	52.3	78.86	91.59	97.24
PV-DBOW	49.27	62.89	79.12	86.82
PV-DM	32.61	61.81	45.96	53.26
PTE	56.62	72.28	91.38	97.74
fastText	60.53	76.01	93.72	97.33
SWEM	66.13	77.64	94.09	96.31
LEAM	61.73	77.99	92.52	94.58
Graph-CNN-C	66.52	77.27	94.05	97.98
Graph-CNN-S	65.39	77.72	93.87	98.18
Graph-CNN-F	66.11	77.96	94.45	97.62
TextGCN	69.34	77.11	94.64	97.79
TextGCN (paper)	70.59	77.82	94.69	97.96
BiHAM	71.28	78.28	94.71	97.93
Textual model	72.59	79.26	96.15	98.43

III. B. Modeling Analysis of Modern Chinese Quantifier Constructions

In this section, the proposed text categorization task is practically applied to analyze the modern Chinese quantifier constructions by taking the quantifier collocations of two nouns with synonymous attributes, “length” and “long and short”, as an example.

III. B. 1) Corpus Sources and Extraction

In this paper, the corpus is extracted from the perspective of attribute values, and there are two positions on the left and right of "length", "length" and measurement phrases: one is that the measurement phrase is on the left, and the format is "the length / length of the measure phrase". One is the measure phrase on the right, in the format "Length/Length X Metric Phrase", where X stands for verb, preposition, or default. Therefore, it is necessary to download the index rows in these two formats and extract the quantifiers that match them for examination. In this paper, the Modern Chinese Corpus (BCC) of Beijing Language and Culture University was selected as the corpus source. The specific steps of corpus extraction are as follows:

(1) Download the index rows, clean and annotate. Taking BCC Modern Chinese as the retrieval scope, 10,000 "length" index rows and 9687 "long and short" index lines were downloaded, respectively, and after cleaning these corpora, they were tokenized and tagged with the help of Python programming technology.

(2) Extract the index rows where "length", "length" and measurement phrases co-occur, and count the quantifiers in them. Firstly, Python is used to check the positions of the third left and right of "length" or "length" in the annotation corpus one by one, and extract the index rows containing quantifiers in the span range. Secondly, after further

screening, the corpus of "length" and "length" with metric phrases was obtained. Finally, the number words and measure words in these corpora are counted and classified, and their similarities and differences are compared.

III. B. 2) Analysis of quantifier constructions based on “length” and “length”

Table 4 shows the results of the statistics and classification of the metric terms paired with "length" and "length". There is some overlap between the quantifiers of "length" and "length", for example, they can be matched with "zhang, ruler, inch, minute, centimeter, inch", which are typical units of length, which verifies their synonym as nouns of measurement attributes. However, there are also very obvious differences between the two, "length" is more frequent than "long" and quantifiers, and the categories of quantifiers are richer and the scope is wider. Specifically, the total frequency of collocation of "length" and quantifiers reached 578 times, of which the collocation frequency with metric length units such as "kilometer and meter" was the highest, accounting for 72.49%, 12.63% with time units such as "day and hour", 6.57% and 3.46% with traditional length units such as "ruler and inch" and imperial length units such as "feet and inches", and 4.84% with proprietary quantifiers in emerging technological fields such as "base pairs" and "bytes". The total frequency of collocation of "length" and quantifiers was 128 times, of which 93.75% were collocated with traditional length units such as "ruler and inch", and there were no more than 4 collocations with metric units, imperial units and time units, and there was no collocation with measurement units in other fields.

Table 4: The corresponding quantifiers and frequencies

Measurement unit		Quantifier	Changduan		Changdu	
			Frequency /times	Proportion /%	Frequency /times	Proportion /%
Time unit		Tian, Ri, year	3	2.34	25	4.33
		Hour, minute, second	0	0.00	48	8.30
		Total	3	2.34	73	12.63
Length unit	Tradition (market)	Zhang	5	3.91	4	0.69
		Chi	58	45.31	21	3.63
		Cun	52	40.62	7	1.21
		Fen	5	3.91	2	0.35
		Li (Huali)	0	0.00	4 (2)	0.69
		Total	120	93.75	38	6.57
	Metric	Kilometer (Gongli)	0	0.00	154 (20)	26.64
		Meter (Gongchi)	0	0.00	168 (6)	29.07
		Centimeter	3	2.34	66	11.42
		Millimeters, micrometers, nanometers	0	0.00	31	5.36
		Total	3	2.34	419	72.49
	British	Feet	0	0.00	11	1.90
		Inch	2	1.56	5	0.87
		Mile	0	0.00	4	0.69
		Total	2	1.56	20	3.46
Other measurement unit		Base pair (bp)	0	0.00	6	1.04
		Byte, character (kb)	0	0.00	14	2.42
		The number of characters, words and letters	0	0.00	8	1.38
		Total	0	0.00	28	4.84
Total			128	100.00	578	100.00

When "length" and "length" are paired with these quantifiers, there are two positions at the same time, but the distribution is quite different. Table 5 shows the left and right position distribution of quantifiers paired with "long". It can be seen that in addition to "centimeter", most of the quantifiers paired with "length" are on the left, and the overall proportion is as high as 93.75%.

Table 5: The left-right position distribution of quantifiers paired with "Changduan"

Quantifier	Year	Zhang	Chi	Cun	Fen	Centimeter	Inch	Total	Proportion /%
Left- position frequency	1	5	58	49	5	0	2	120	93.75
Right-position frequency	2	0	0	3	0	3	0	8	6.25
Total	3	5	58	52	5	3	2	128	100.00

Table 6 shows the left-right position distribution of quantifiers paired with "length". It can be seen that in addition to the categories of "zhang, ruler, inch, and divide", most of the quantifiers matched with "length" are on the right, and the overall proportion reaches 84.26%. In addition, as previously researched and said, the numerical range of numbers paired with "length" is large, the smallest is "a few tenths of a few", the largest is "hundreds of millions", and most of them are written in Arabic numerals, with many decimal forms. The numerical range with "long and short" is narrow, the largest is not more than "thousand", the smallest is "one", and most of them are integers from "one" to "ten" written in the form of Chinese characters, and more include "several, dozens, remainder, number, dozens" and other forms of approximate expressions, and there are no decimal or fractional forms.

Table 6: The left-right position distribution of quantifiers paired with "Changdu"

Quantifier-changdu	Left- position frequency	Proportion /%	Right-position frequency	Proportion /%	Total
Tian, Ri, year, minute, second	30	41.10	43	58.90	73
Zhang, Chi, Cun, Fen, Li (Huali)	24	63.16	14	36.84	38
Kilometer (Gongli)	8	5.19	146	94.81	154
Meter (Gongchi)	13	7.74	155	92.26	168
Centimeter	11	16.67	55	83.33	66
Millimeters, micrometers, nanometers	2	6.45	29	93.55	31
Feet, inch, mile	0	0.00	20	100.00	20
Base pair (bp)	0	0.00	6	100.00	6
Byte, character (kb)	2	14.29	12	85.71	14
The number of characters, words and letters	1	12.50	7	87.50	8
Total	91	15.74	487	84.26	578

Based on the above findings, the classical construction of the quantifier of “length” can be summarized as follows:
Length + (X) + number word [0.00~hundreds of millions] + measure word [kilometers/meter/centimeters].
The classical construction of the quantity word for “length” is:
Number word [one to a thousand] + Quantity word [feet/inches] + Length
A noun of an attribute paired with a quantifier indicates that the attribute is measurable, and can be characterized semantically as [+measure]. From the above, we can see that both “length” and “length” have the semantic feature of [+measureability], but the former is more significant in this feature and is a typical attribute noun of measurement. The latter's [+metricity] semantic feature needs to be presented under certain conditions, and is less salient, making it a restricted metric attribute noun.

IV. Conclusion

In this paper, we realized the modeling and analysis of modern Chinese quantifier constructions by constructing a multi-label text classification model of modern Chinese with fused graph convolutional networks.

On the four experimental datasets of Ohsumed, MR, R52, and R8, the classification accuracies of the model after BiGRU is added to GCN are improved by 1.39%, 1.66%, 1.65%, 1.56%, respectively, and the classification accuracies of the model after BERT is added to GCN are improved by 1.72%, 1.22%, 1.87%, 2.29%, respectively, and the classification accuracies of the model of this paper's BERT + GCN model's classification accuracy is significantly better than that of BiGRU+GCN model, which proves the feasibility of combining BERT model with graph convolutional neural network for modern Chinese text classification. Meanwhile, the classification effect of this paper's model on the four datasets achieves the best among all the compared models, reaching 72.59%, 79.26%, 96.15%, and 98.43% on the four datasets of Ohsumed, MR, R52, and R8, respectively, which is an improvement of 1.31%, 0.98%, 1.44%, and 0.50% compared with that of BiHAM model, respectively.

The model in this paper is used to model the attribute word construction modeling of the synonymous attribute nouns "length" and "length", and the similarities are as follows: both can be matched with quantifiers, and there are some common quantifiers, such as "zhang, ruler, inch, minute, centimeter, inch", etc., and there are two positions in the collocation with quantifiers. This verifies that "length" and "length" are synonymous, both attribute nouns, and

are metric. The difference is that the frequency of co-occurrence of "length" and quantifiers is high, the number of collocation quantifiers is large, the categories are rich and the range is wide, and the numerical range of collocations is large and the forms are rich. The frequency of "length" and quantifiers is low, and the number of collocation quantifiers is small, and the categories are mainly traditional length units such as "ruler and inch". The numerical range of collocations is small, and most of them are integers.

References

[1] KHASANOVA, F. (2022). The Formation and Formation of The Chinese Language Baihua as The Basis of The Modern Chinese Language. SHARQ MASH'ALI, (01), 21-23.

[2] Ross, C., Ma, J. H. S., Chen, P. C., He, B., & Yeh, M. (2024). Modern Mandarin Chinese grammar: A practical guide. Routledge.

[3] Frajzyngier, Z., Liu, M., & Ye, Y. (2020). Reference system in modern Mandarin Chinese. Australian Journal of Linguistics, 40(1), 45-73.

[4] Kolpachkova, E. (2021). Causal Constructions in Modern Chinese. Confucius Institute in Sofia, 2021, 25.

[5] Weiguo, S. I., & Xu, W. E. N. (2024). The Motivations of the Modern Chinese Verb-Copying Construction. Journal of Foreign Languages, 47(3), 53-62.

[6] Racine, J. P. (2018). Lexical approach. The TESOL encyclopedia of English language teaching, 2, 1-7.

[7] Müller, S., & Wechsler, S. (2014). Lexical approaches to argument structure. Theoretical Linguistics, 40(1-2), 1-76.

[8] Zhu, X., Liao, X., & Cheong, C. M. (2019). Strategy use in oral communication with competent synthesis and complex interaction. Journal of psycholinguistic research, 48, 1163-1183.

[9] Qiu, L., Lu, J., Ramsay, J., Yang, S., Qu, W., & Zhu, T. (2017). Personality expression in Chinese language use. International Journal of Psychology, 52(6), 463-472.

[10] Chen, X. (2017). Extensions of the Chinese passive construction. East Asian Pragmatics, 2(1), 59-74.

[11] Yang, X., & Wu, Y. (2020). On the scope of quantifier phrases in Chinese passive construction. International Journal of Chinese Linguistics, 7(1), 71-89.

[12] Xu, J. (2025). A Study of Modern Chinese Quantifier Constructions Based on Random Forest (RF) Modeling. J. COMBIN. MATH. COMBIN. COMPUT, 127, 55-63.

[13] Wu, K. (2018, January). The Study of Quantifiers in Teaching Chinese as a Foreign Language. In 2017 5th International Education, Economics, Social Science, Arts, Sports and Management Engineering Conference (IEESASM 2017) (pp. 182-185). Atlantis Press.

[14] Chen, Z., & Shao, B. (2024). Alternation in the Chinese Event-quantifying Construction: A multivariate approach. Lingua, 305, 103741.

[15] Pan, X., & Liu, H. (2014). Adnominal Constructions in Modern Chinese and their Distribution Properties. Glottometrics, 29, 1-30.

[16] Crain, S. (2017). Acquisition of quantifiers. Annual Review of Linguistics, 3(1), 219-243.

[17] Khushnood Abbas, Shi Dong, Alireza Abbasi & Yong Tang. (2025). Cross-domain inductive applications with unsupervised (dynamic) Graph Neural Networks (GNN): Leveraging Siamese GNN and energy-based PMI optimization. Physica D: Nonlinear Phenomena,476,134632-134632.

[18] Ilias Kalouptsoglou, Miltiadis Siavvas, Apostolos Ampatzoglou, Dionysios Kehagias & Alexander Chatzigeorgiou. (2025). Transfer learning for software vulnerability prediction using Transformer models. The Journal of Systems & Software,227,112448-112448.

[19] Gaoqing Xu, Qun Chen, Shuhang Jiang, Xiaohang Fu, Yiwei Wang & Qingchun Jiao. (2025). Analyzing the capability description of testing institution in Chinese phrase using a joint approach of semi-supervised K-Means clustering and BERT. Scientific Reports,15(1),11331-11331.

[20] Yingjie Du, Ning Ding & Hongyu Lv. (2025). Spatio-temporal prediction of terrorist attacks based on GCN-LSTM. Journal of Safety Science and Resilience,6(2),186-195.