International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 4328-4341

https://doi.org/10.70517/ijhsa464280

A Study of Graph Neural Network-based Text Representation Learning Methods in English Sentiment Analysis

Xiao Liu[1,*]
[1] Department of Basic Science, Shaanxi University of International Trade & Commerce, Xi'an, Shaanxi, 712046, China
Corresponding authors: (e-mail: liuxiao001485@163.com).

Abstract With the rapid development of communication and computer technology, Internet applications continue to penetrate into all aspects of society, and text, as a carrier for people to directly express their emotions and opinions, occupies a large proportion of network data. In this paper, we choose the comment text as the experimental data and sentiment analysis as the research task, and design a text sentiment analysis model based on graph neural network and representation learning. English is taken as the source language, and six languages, Chinese, French, German, Japanese, Korean and Thai, are respectively taken as the target languages for analysis. The experimental results show that the sentiment analysis model proposed in this paper improves the F1 value of sentiment classification in cross-language environment more, and then compares with the machine translation method and the cross-language sentiment analysis method without sentiment feature representation, which improves the prediction accuracy by about 10.21% and 9.7%, respectively, and has a better performance in text sentiment.

Index Terms text representation, text sentiment analysis, graph neural network, cross-lingual

I. Introduction

English sentiment analysis, that is, the process of analyzing, processing, summarizing and then reasoning through subjective English text with emotional color [1], [2]. With the arrival of the big data era, there is an increasing demand for processing and analyzing English text data, and graph neural network representation learning, as an important research direction in the field of artificial intelligence, plays an important role in solving the problems of complex relationships, semantic understanding and sentiment analysis of English text [3]-[6].

Representation learning is a method to convert complex graph data structures into low-dimensional vectors in order to better mine the information in the graphs and thus improve the efficiency of downstream tasks [7], [8]. In recent years, data in areas such as social networks, transportation networks and bioinformatics have become increasingly prevalent, and effective graph representation learning methods are needed to process these data in order to solve real-world problems [9]-[11]. The core of representation learning is to extract useful information from data by learning an effective representation of the data, which can be either low-dimensional dense vectors or high-dimensional sparse vectors, with the aim of capturing the intrinsic structure and relationships among the data, where the most common assumption is that the data has a low-dimensional manifold structure, i.e., the data is distributed on a low-dimensional manifold [12]-[15]. By mapping the data onto this low-dimensional manifold, data downscaling and feature extraction can be achieved to better understand the data [16], [17]. Traditional methods of representation learning often can only capture the surface features of text, lack of generalization ability in different domains or contexts, difficult to deal with long text as well as semantic ambiguity, and often ignore the complex relationship between words and the sequential features of the text when dealing with text, which limits the accuracy and depth of the sentiment analysis [18]-[21]. In contrast, the text representation learning method based on graph neural networks effectively reveals the semantic relationships between texts by constructing the relationships between them, and has achieved impressive results in English sentiment analysis [22]-[24].

In order to solve the shortcomings of traditional text sentiment analysis in different domains or contexts, such as the lack of generalization ability and the pooling layer destroying the text sequence features, this paper utilizes graph convolutional neural network for coding, and proposes a text sentiment analysis model based on graph neural network with representation learning. Relevant datasets are selected as experimental data, and English is used as the source language, and six different languages from three datasets, Chinese, French, German, Japanese, Korean, and Thai, are tested as the target language machine translation method, and the cross-language sentiment analysis method that does not use the representation of sentiment features. In order to verify the generalization of the model on aspect-level sentiment analysis tasks, a comparative analysis is performed on the REST15 and REST16 datasets.

II. English Text Representation and Sentiment Analysis Model Based on Graph Neural Networks

II. A. Graph Neural Networks

Graph neural networks are mainly used for information extraction and analysis of graph-structured data. Early graph neural networks as a generalization of recurrent neural networks are computationally complex and inefficient. With the popularity of convolutional neural networks in the field of deep learning, many researchers hope to apply convolution to graph neural networks, and the current mainstream methods are divided into two categories: spectral domain and null domain, the former learns node features from the perspective of spectral signals, and the latter extracts the node information directly from the graph structure.

Graph Convolutional Neural Network (GCN), as a classical spectral domain method, constructs a hierarchical linear model on the basis of spectral domain graph convolution, extracts the features by using multi-layer convolution, and then simplifies the parameters by approximation, which greatly reduces the amount of computation, and realizes the learning and classification of graph node feature representation [25].

Firstly, we introduce the spectral domain graph convolution, which is the theoretical foundation of GCN; then we introduce the hierarchical linear model, which enables the graph convolution to stack multiple layers as CNN; finally, we introduce the GCN realization formula as well as the objective function.

The spectral domain convolution on the graph is defined as the product of the signal $x \in \mathbb{R}^N$ and the filter $g_\theta = diag(\theta)$ parameterized by the parameter $\theta \in \mathbb{R}^N$ in the Fourier domain:

$$g_\theta * x = U g_\theta U^T x \tag{1}$$

where U is the eigenvector matrix of the normalized graph Laplacian matrix $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$, I_N is the N th order unitary matrix, Λ is the diagonal matrix of eigenvalues, $U^T x$ is the graph Fourier transform of x, and here g_θ is used as an eigenvalue function for L in the form $g_\theta(\Lambda)$.

Since the product of eigenvalue matrices is computationally expensive with a time complexity of $O(n^2)$, Eq. (1) is computationally expensive. And the matrix eigen-decomposition of L may also be very time-consuming for large graphs.

To get around this problem, for $g_\theta(\Lambda)$, it is approximated by a truncated expansion of the Chebyshev polynomial $T_k(x)$ up to order K as follows:

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda}) \tag{2}$$

Substituting this into Eq. (1) as g_θ gives:

$$g_{\theta'} * x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L}) x \tag{3}$$

where $\tilde{L} = \frac{2}{\lambda_{\max}} L - I_N$, and $\lambda_{\max}$ is approximated by 2. Notice that this formula is K-localized, since it is a Laplace's k-order expansion that depends only on the node's K-order neighborhood of the of the node.

Referring to the multi-layer convolution stacking process of CNN, GCN tries to build a graph convolution-based neural network model by stacking multiple convolutional layers as in the form of Eq. (3), with a nonlinear activation function added after each layer. When $K = 1$ in Equation (3), the model becomes a linear function in the spectral domain.

Instead of $\tilde{L}$, a symmetrically normalized normalized Laplace matrix is used, and in order to simplify the computational effort, let $K = 1$, which further simplifies Eq. (3) to:

$$g_{\theta'} * x \approx \theta'_0 x + \theta'_1 (L - I_N) x = \theta(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}) x \tag{4}$$

where $\theta = \theta'_0 = -\theta'_1, \tilde{A} = A + I_N, \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

Equation (4) can be used as a filter on the spectral signal x on the graph, to which the multilayer convolution described above is applied, thus enabling the convolution of the k th order neighborhood of the node, where k is the number of convolutional layers in the neural network model.

Converted to matrix form, and then adding the activation function after the current layer, the node features after convolution of the current layer can be obtained:

$$Z = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW) \tag{5}$$

where σ is the ReLU activation function, $X \in \mathbb{R}^{N} \times C, W \in \mathbb{R}^{C\times M}$, and the final post-convolutional node features $Z \in \mathbb{R}^{N\times M}$.

GCN is by default oriented to semi-supervised tasks is to take all the features as input, but only part of the labeling is used as supervised information, and the loss function is to evaluate the cross-entropy loss of all labeled samples:

$$Loss_{gcn} = -\sum_{l\in y_c}\sum_{m=1}^{M} Y_{lm} \ln Z_{lm} \tag{6}$$

where Z_{lm} is the classification result obtained by the loss obtained from the final hidden layer and then softmaxed, Y_{lm} is the true label, and y_c is the set of nodes in the fraction with labels as supervised.

II. B. Text representation

II. B. 1) Vector space model

Vector space models have been widely used in information retrieval and natural language processing tasks and are often also referred to as bag-of-words models. The idea of vector space modeling is to represent document semantics through the words it uses, using words as vector attributes, and composing vectors by calculating the word contribution as the attribute value [26].

In the space vector model, a document is represented as a sequence of words, which is formalized as d_i for a document d_i with index number i, where $d_i = (w_{i,1}, w_{i,2}, ..., w_{i,m})$, where m is the document length. Usually a document is a piece of text, or a localized paragraph, sentence, etc. of text. In the retrieval task, the query is the content to be found, which is also represented as a sequence of words $q = (w_{q,1}, w_{q,2}, ..., w_{q,t})$, where t is the query length. In order to map documents and queries into numeric vectors, it is necessary to determine the set of feature terms (Terms), which usually refers to the set T of basic linguistic units such as words, phrases, word groups, pieces of language, etc. The feature items are used as attributes to form the vector space, and the attribute values of the feature items in documents and queries are computed, followed by mapping the documents and queries to the feature space with the spatial dimension $|T|$. The commonly used methods for computing attribute values are:

Term Frequency (TF), the number of times the feature item appears in the document.

Document Frequency (DF), the ratio of the number of document pages containing the feature item to the total number of document pages in the entire dataset;

Inverse Document Frequency (IDF), the inverse of document frequency, which reflects the degree of imbalance in the distribution of the feature term in the dataset.

Document weight (TF-IDF), which integrates document frequency and inverse document frequency.

In text sentiment analysis, Boolean weights are also often used, i.e., whether the feature word appears in the document or not, if it appears take the value 1, otherwise take the value 0.

In retrieval tasks, cosine values are often used as a similarity measure between the document and the query. Let $V_T(x)$ be a numerical vector representation of a text sequence x over a set of feature terms T, then the cosine similarity measure of document d_i and query q is as follows.

$$Cosine(V_T(d_i), V_T(q)) = \frac{V_T(d_i)V_T(q)}{|V_T(d_i)| \times |V_T(q)|} \tag{7}$$

The vector space model is simple to construct, based on the foundation of linear algebra, which is easy to be used in combination with existing machine learning methods and models. Moreover, the vector weights are easy to calculate, and the text vectorization process is flexible, which can be matched locally or retrieved globally. However, when the length of the document is too long, it contains too many words, and the query accuracy will be reduced. The more important defect is that the vector space model does not consider the semantic associations between words, and it is mandatory that the feature items are semantically independent, and at the same time, it needs to be matched precisely in order to map the word sequences into the vector space, and the representation dimension is high and sparse, and the ability to represent the association between the semantics is poor. Even so, the vector space model has achieved better performance in traditional text categorization tasks, and has also been widely used in text sentiment analysis tasks.

II. B. 2) Fisher feature selection

Chapter sentiment polarity classification uses supervised classification methods that require extraction of classification features from training data. It has been shown that feature selection methods such as information gain, chi-square statistic, and log-likelihood ratio, which are often used in traditional text classification methods, can be used for sentiment feature selection. Aiming at the characteristics of emotional features, a Fisher linear discriminant ratio based emotional feature selection method is proposed. The method is able to select features with large category differentiation and good data hugging, and thus can effectively improve the classification performance. The sentiment feature selection method based on Fisher linear discriminant ratio is divided into the following steps.

(1) The text dataset is preprocessed by a preprocessing process, and all the words contained in it are taken as the candidate feature set T.

(2) All the texts in the dataset are transformed into a spatial vector representation on the candidate feature set T, labeled as matrix χ, where the attribute value calculation method can be selected according to the demand.

(3) In χ, each row represents a sample, and each column is the value of the corresponding candidate feature over all samples. Based on the sample labels of the training set, the sample expectation and sample variance of the values of each candidate feature in each column of χ in the positive and negative categories are computed respectively, labeled as $E_{pos}[t], E_{neg}[t], Var_{pos}[t]$ and $Var_{neg}[t]$, where $t \in T$ is the candidate feature. The candidate features are scored according to the definition of Fisher's linear discriminant ratio, and for candidate feature t, the score is as follows.

$$J_F(t) = \frac{(E_{pos}[t] - E_{neg}[t])^2}{Var_{pos}[t] + Var_{neg}[t]} \tag{8}$$

(4) Arrange all the candidate features in descending order according to their scores, and select the part with high scores as the representation features. After removing the non-feature columns from χ, the remaining feature columns form a matrix that is the data representation after feature selection.

After feature selection, the data set is converted into a smaller data representation matrix. After that, the data matrix is used as input to the classification model and the classifier is trained to be used for the text sentiment prediction task.

II. C. English Text Feature Extraction

II. C. 1) Building a feature dictionary

The core of constructing a feature dictionary of English text summaries is to select feature words that can represent certain subordinate features in the cloud environment and have the significance of summarizing and guiding. The selection of feature words is based on the outline content of the English text summaries, and the use of concise word frequency maximizes the overview of the overall information of the English text summaries. Usually, the feature words meet two requirements, one is to be able to use bag-of-words model to describe the feature words and successfully complete the word frequency weighting strategy, and the other is that the feature word EEC value meets the measurement range.

The word frequency is divided into two categories, A and B. Bringing the feature words of text w_i into the two categories respectively, there will be the phenomenon that the feature word weight value tends to favor the A category, equivalent to the exclusion of B. This is due to the difference of the feature word weight value in the cloud computing environment, and the expression of the feature word weight value of text w_i is as follows:

$$ECE(w_i, C_j) = P(w_i \mid C_j) P(C_j \mid w_i) \lg_2 P(C_j \mid w_i) / P(C_j) \tag{9}$$

where text w_i is negatively correlated with category C_j, i.e., the effect of text w_i on category C_j when the relative value of $P(w_i \mid C_j)$ is small and the relative value of $P(C_j \mid w_i)$ is large is Neglecting this, the weight value of text w_i in the stochastic categorization $ECE(w_i, C_j)$ can be derived from $P(C_j)$.

The functional equation for the value of $ECE^{'}$ for textual data in category C_j is as follows:

$$ECE^{'}(w_i, C_j) = -\sum_{k \neq j} ECE(w_i, C_j) + ECE(w_i, C_j) \tag{10}$$

where $ECE(w_i, C_j)$ denotes the weight value of category C_j; $-\sum_{k=i} ECE(w_i, C_j)$ denotes the end of textual w_i full data weighting when $h \neq j$, indicating the categorized category.

According to the $ECE^{'}$ value of the text data can calculate the feature words of any category C_j, the expression is as follows:

$$w_{i,j} = ECE^{'}(w_i, C_j)tl \tag{11}$$

where l denotes the English text summary coefficient; t denotes the feature word parameter. When solving the feature words, it is necessary to compare the $w_{i,j}$ values and arrange them neatly in the order of size, parameterize the t feature word of the first sequence as the feature word of the text w_i, and repeat the operation on the text data until all feature words are obtained. The obtained feature words of all categories are integrated to form a feature dictionary together.

II. C. 2) Selection of the optimal theme

Selecting the optimal theme is the core of the above operation, and the weighting strategy is imported into the theme to determine the ability of each category according to the change of the weighting value. The higher the weight value, the stronger the ability of the corresponding theme's category.

In the process of optimal topic selection, firstly, the number of topics in the set N is counted, and according to the categories, all topics are distinguished into small groups that are not easy to be confused, and different elements are selected as the category representatives of each group, and all the category representatives in the set N build the knowledge background model together, and are able to obtain $|N|$ categories of topics, which are represented as $S = \{S_l, S_2, \cdots, S_{|N|}\}$.

The theme weight vector determines the differentiation ability of different categories and its expression is as follows:

$$W = w(S_1), w(S_2), \cdots, w(S_{|N|}) \tag{12}$$

where $w(S_{|N|})$ denotes the importance degree of the topic $S_{|N|}$, and the importance degree of the topic $S_{|N|}$ gradually decreases with the gradual decrease of $w(S_{|N|})$.

In order to strengthen the classification accuracy, the topic with outstanding performance of weight values among the $|N|$ topics is considered as the optimal topic. The Relidf feature optimization idea is adopted to compare the weight values of all the texts in the background knowledge model, and the function expression is as follows:

$$\begin{aligned} w(S_i) = &\sum_{i=1}^{n} dis[d_i^{S_i}, nm(d_i)^{S_i}] + w_{i,j} \\ &-\sum_{i=1}^{n} dis[d_i^{S_i}, nh(d_i)^{S_i}] \end{aligned} \tag{13}$$

where $dis[d_i^{S_i}, nh(d_i)^{S_i}]$ denotes the difference in the weight value between text i and text j for any topic. In the topic S_i, the smaller the difference between two texts with similar types, the larger the difference between two texts with different types. When the topic weight value exhibits $S_i = \arg\max_{S_i \in S}[w(S)]$, the topic can be regarded as the optimal topic.

II. C. 3) Generating new text features

The generation of new text features is based on the above analysis, and the category distribution of the corresponding topics is performed by Gibbs sampling, combining the text d_i and the optimal topic S_i to obtain the category distribution result $d_i^{S_i}$. Substitute the category distribution result $d_i^{S_i}$ in the feature dictionary to obtain new features that can describe the text d_i with the following functional expression:

$$F_d = [w_d, \alpha d_i^{S_i}] \tag{14}$$

where α denotes the weight value of the subject feature and w_d denotes the vector value of the feature dictionary.

II. D. Sentiment analysis model based on graph neural network representation learning

The text sentiment analysis model (GNNRL) proposed in this paper consists of a word embedding layer, a graph convolution layer, a pooling layer, and an output layer, and its structure is shown in Fig. 1.

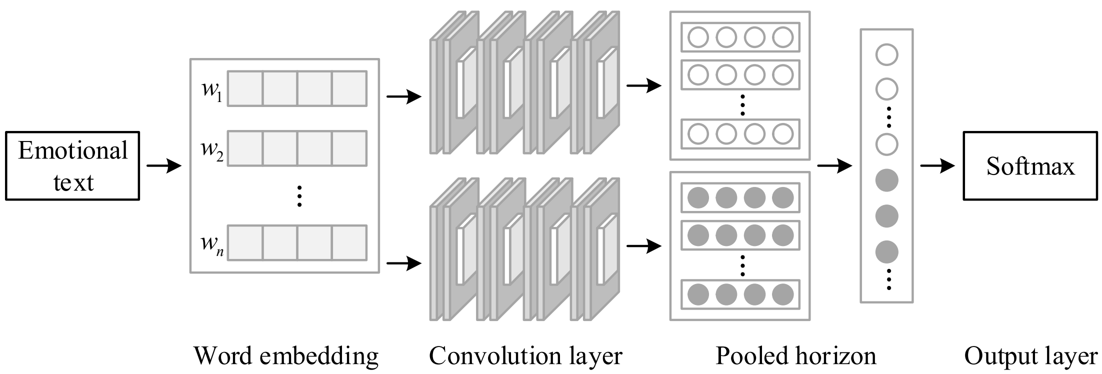


Figure 1: Model structure drawing

II. D. 1) Word Embedding Layer

Text analysis requires the conversion of raw text data into vector representations, and this paper draws on a word vector model based on large-scale corpus pre-training proposed in the literature, which is able to consider all positions when processing input sequences and assigns a different attentional weight to each position, enabling the model to more accurately represent the relationships between words, sentences, and contexts.

II. D. 2) Graph Convolution Layer

The traditional convolution operation has a problem in extracting n-grams features, that is, the length of the extracted feature information is reduced compared to the original information. In this paper, the extra extracted features using wide convolution are mainly focused on the two ends of the text, which makes the extracted features more comprehensive and helps to improve the expressiveness of the features.

The syntactic dependency tree of a sentence is generated using the Spacy dependency parser, a structure that demonstrates the dependencies between words in a sentence. In this process, aspect words are labeled in blue font, emotion words are labeled in red font, edges denote dependencies, and labels denote the type of dependency, consisting of syntactic dependency labeling words such as det, acomp, scounj, nsubj, advmod, and marker.

For example, for a text containing n words, in this paper, we use $w_i \in V^d$ to denote the d-dimensional word vectors w_i obtained by GloVe word vector embedding for the i th word of the word. These word vectors are then combined into a text matrix T, denoted as $T = [w_1, w_2, \cdots, w_n]$.

In constructing a multifaceted sentiment map, this paper considers the dependency information between aspects of a sentence and encodes a syntactic dependency tree using graph convolutional neural networks. The syntactic dependency tree can be regarded as a contextual syntactic graph $G_{xy}(V, E)$, where the number of nodes in the multifaceted sentiment graph is determined by the number of aspects, and the node V denotes the hidden state of a word while the edge E denotes the connection between words. The relationship between each node v_i and v_j can be represented by the mathematical expression (15).

$$A_{ij}^{vy} = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where A_{ij}^{xy} is the adjacency matrix, and $i, j \in [1, n], e_{ij} \in E$ denotes that node i depends on j.

In order to consider the relative distance between context and aspect words, positional coding is introduced in this paper. As shown in Eq. (16), the positional encoding is defined using the index of the aspect word and the relative distance of the context word, thus effectively capturing the syntactic relationship between the aspect word and the context.

$$p_i = \begin{cases} 1 - \dfrac{a_{start} - i}{n} & 0 \le i < a_{start} \\ 0 & a_{start} \le i < a_{end} \\ 1 - \dfrac{i - a_{end}}{n} & a_{end} \le i \le n \end{cases} \tag{16}$$

where a_{start} is the start index and end index, n is the sentence length, and p_i is the relative distance from the context word to the aspect.

The graph convolution layer effectively captures the relationship between the nodes through information transfer and aggregation between the graph nodes and continuously updates the representation of the nodes in a multilayer structure, in each layer, the GCN aggregates the contextual information for better understanding of the syntactic structure in the sentence, and the graph convolution operation of the normalization factor is shown in equation (17).

$$h_i^{sy(l)} = RELU\left(\sum_{j=1}^{n} \tilde{A}_{ij}^{xy} h_j^{l-1} W^l + b^l \right) \tag{17}$$

where $\tilde{A}^{xy} = \tilde{D}^{-\frac{1}{2}} A^{xy} \tilde{D}^{1/2}$ is the normalized symmetric matrix, $\tilde{D}$ is the degree matrix of A^{xy}, h_j^{l-1} is the upper layer positional word representation, and W^l and b^l are the weight matrix and bias of the l th layer.

II. D. 3) Pooling layer

The features are further filtered after the convolution operation. Compared with maximum pooling, k-max pooling serves to extract the k most important feature values from the text, and this method effectively preserves the sequential features of the textual information, thus avoiding the loss of information. Instead, the dynamic adjustment of k values is based on the length of the input text and the number of layers of the current network. At shallower levels of the network, more feature values need to be extracted to support subsequent processing steps, so a relatively large k value is chosen; while at deeper levels of the network, the k value is gradually reduced to preserve important features while reducing the number of network parameters. This dynamic change makes the network structure more flexible. The way of choosing k-value can be expressed by the following equation: the k-value is chosen as shown in equation (18).

$$k_l = \max(k_{top}, \frac{L - l}{L} n) \tag{18}$$

where L is the total number of layers in the convolutional layer; l denotes the number of layers where the current pooling operation is located; k_{top} is the fixed pooling k value of the top layer; and n denotes the length of the input sentence.

The convolutional and pooling layers form a feature extraction layer in concert. Let denote the feature extracted by the j th convolutional kernel of the i th layer, then it can be expressed in the following form:

$$D_j^i = \sum_{k=1}^{n} m_{j,k}^j * D_k^{i-1} \tag{19}$$

where $m_{j,k}^{j}$ denotes the parameter matrix and D_k^{i-1} denotes the features of the k th convolution kernel of the $i-1$ th layer. After the above computational processing, the final output is shown in Eq. (20) and Eq. (21).

$$D^{top}=[D_1^{top},D_2^{top},\cdots,D_n^{top}] \tag{20}$$

$$D_j^{top}=\sum_{k=1}^{n} m_{j.k}^{top*}D_k^{top-1} \tag{21}$$

II. D. 4) Output layer

Finally, the sentiment feature vector D is transported to the classifier SoftMax to determine the sentiment classification based on the normalized value.SoftMax is a logistic regression technique used in multiple classifiers, which makes the category classification more effective and precise by summing up the features that can be identified as the same category and converting them to a probability value. It can help us to better understand and predict the relationship between categories, which is calculated as shown in equation (22).

$$p(y=i)=\frac{e^{XW_i^T}}{\sum_{j=0}^{k-1}e^{XW_j^T+b_j}} \tag{22}$$

where the text feature vector is denoted by X, k is the number of categories, and the corresponding loss function is shown in equation (23).

$$L(\theta)=-\log p(y=i) \tag{23}$$

III. Experimentation and analysis

III. A. Experimental setup

III. A. 1) Experimental data

In this paper, we conduct experiments using a total of 100,000 microblog text data nCoV 100k labled publicly available on the DataFountain competition platform, which has been labeled manually and classified into 3 categories: 1 (positive), 0 (neutral) and -1 (negative).

In order for the model to be better trained, the data needs to be preprocessed. Firstly, the Pandas toolkit in Python is used to process the data with outliers, to eliminate the data whose sentiment classification does not belong to 1(positive), 0(neutral) and -1(negative), and then the data is subjected to operations such as de-weighting and elimination of meaningless symbols.

98,813 data remained after preprocessing and the sentiment labels were modified to 0 (negative), 1 (neutral) and 2 (positive). All data were cut into training set, validation set, and test set in the ratio 6:2:2. The sample distribution of the training set is shown in Fig. 2 and the text length distribution is shown in Fig. 3.

As can be seen from Fig. 2, the sample categories in the training set are not balanced, among which, the 1(neutral) category accounts for the largest proportion. As a whole, roughly 55% have a neutral attitude, 35% have a positive attitude, and about 10% have a negative attitude. Therefore, the effect of category imbalance on the test results needs to be taken into account when testing the model.

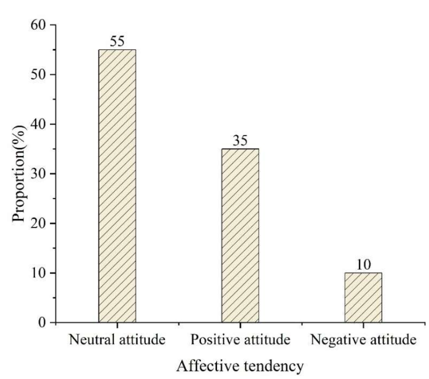


Figure 2: Distribution of samples in the training set

As can be seen from Figure 3, the length of the text in the training set is almost between 0 and 160 words, so when setting the word embedding dimension in the model training, it can be set to 160 to avoid the waste of space under the premise of ensuring the authenticity of the information.

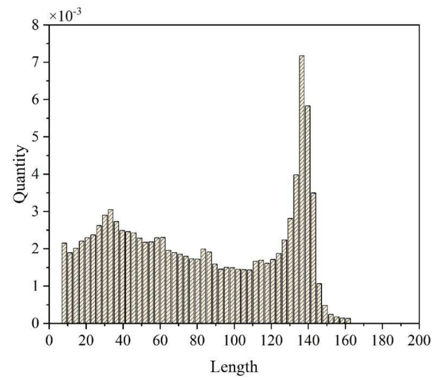


Figure 3: Text length distribution

III. A. 2) Experimental environment

The experimental environment is shown in Table 1:

Table 1: The experimental environment

Experimental environment	Experimental configuration
Operating system	Windows10
Development tool	Pycharm
Development language	Python3.9
Depth learning framework	Pytorch-gpu-1.7.1

III. B. Text representation learning experiment analysis

III. B. 1) Acquisition of a priori features

Taking the EB-CB transfer learning task as an example, Table 2 shows the representation of some terms in English and Chinese documents. Among them, "fun", "masterpiece", "disappointed" and "praise" are the core words selected on the source language training set, which are translated into Chinese with the help of one-to-one translation, and DCF is used to calculate the correlation between the core terms and all the terms on the unlabeled documents of the two languages, and each term is represented as a vector composed of M core word items. For the core words, the term representations of the source domain and the target domain are calculated on the unlabeled corpus, and the average value of the two is used as the unified result, so that the core words have the same term representation in the source domain and the target domain. As can be seen from the table, the core terms "book" (source domain) and "book" (target domain) have the same term representation (they have the same relevance values as fun, masterpiece, dispointed, and praise). The word "good-looking" expresses positive emotions, which are positively correlated with "fun", "masterpieces", and "praise" with positive emotions, and negatively correlated with negative emotions "disappointment", indicating the validity of the semantics of textual representation.

Table 2: Examples of term profiles

Term term	Core word			
	Fun	Masterpiece	Disappointed	Praise
Book	-0.000918	-0.001123	-0.007317	0.002619
Good	0.078615	0.031137	-0.017695	0.071537
Favorite	0.317367	0.612792	0.244357	0.148979
书(Chinese)	-0.000987	-0.001129	-0.007445	0.002719
喜欢(Chinese)	0.223081	0.314409	-0.321758	0.932288
最爱(Chinese)	0.354517	0.399587	-0.612236	0.394569

III. B. 2) Cross-language same-domain experiments

In the same domain and cross-language experiments, Upper, MT, BiLDA and DCI are selected as the comparison methods, and English is taken as the source language and other languages as the target language. Comparing the document classification F1 values of each method in cross-language environment, the results are shown in Table 3.

The GNNRL proposed in this paper achieves better F1 values than DCI in the sets of experiments EM-JM, EB-CB, ED-CD, and EM-CM, with the results of 0.6459, 0.6937, 0.7729, and 0.7404, respectively.For the BiLDA method, although it extracts the common topic distribution in the same domains of the two languages, this topic distribution may not be applicable to sentiment categorization, resulting in a low F1 value for its classification.

Table 3: The results under cross language environment

$L_s D_s - L_t D_t$	Upper	MT	BiLDA	DCI	GNNRL
EB-JB	0.7323	0.6384	0.4875	0.7244	0.6016
ED-JD	0.7575	0.6673	0.4849	0.7196	0.5994
EM-JM	0.7779	0.6642	0.5009	0.5892	0.6381
EB-CB	0.7822	0.6613	0.5059	0.5284	0.6859
ED-CD	0.7252	0.6439	0.4631	0.5353	0.7651
EM-CM	0.7627	0.6352	0.4723	0.5125	0.7326

III. B. 3) Cross-language cross-domain experiments

The experiments in the cross-lingual and cross-domain environment are more complex than the previous cross-lingual and same-domain experiments. In this set of experiments, there is no common feature intersection between documents due to the different languages used to represent the documents; the semantics expressed by the comments will also change somewhat due to the different domains. The results of their experiments are shown in Table 4.

The model proposed in this paper uses CHI instead of MI for computation when using the DCI method to select core word pairs between different domains and languages. In the cross-language and cross-domain environment, GNNRL has further improved than DCI and MT in sentiment classification of Chinese comments, which is mainly due to the following reasons: (1) the change of the calculation method of core word pairs selection, after replacing the feature selection method from MI to CHI, the number of core feature pairs acquired increases, and more core features are retained; (2) encoding and dimensionality reduction of the core features, which makes the network can extract more effective features for classification in cross-linguistic and domain environments based on the original TP features; (3) Chinese comments are shorter in length compared to Japanese and English comments, and the text length may be one of the reasons affecting the effect of GNNRL.

Table 4: The results under cross language/domain environment

$L_s D_s - L_t D_t$	MT	DCI	GNNRL	$L_s D_s - L_t D_t$	MT	DCI	GNNRL
EB-JB	0.6430	0.7242	0.7089	EB-JB	0.6353	0.6855	0.6160
ED-JD	0.6261	0.7704	0.6697	ED-JD	0.6139	0.593-	0.575
EM-JM	0.6298	0.4999	0.5244	EM-JM	0.6581	0.5237	0.5932
EB-CB	0.5945	0.4954	0.7077	EB-CB	0.6512	0.5086	0.7282
ED-CD	0.6336	0.5069	0.6838	ED-CD	0.6354	0.4995	0.7379
EM-CM	0.6801	0.497	0.7418	EM-CM	0.6205	0.5069	0.7781
EB-JB	0.6430	0.7242	0.7089	EB-JB	0.6353	0.6855	0.6160

III. C. Text Sentiment Analysis Results

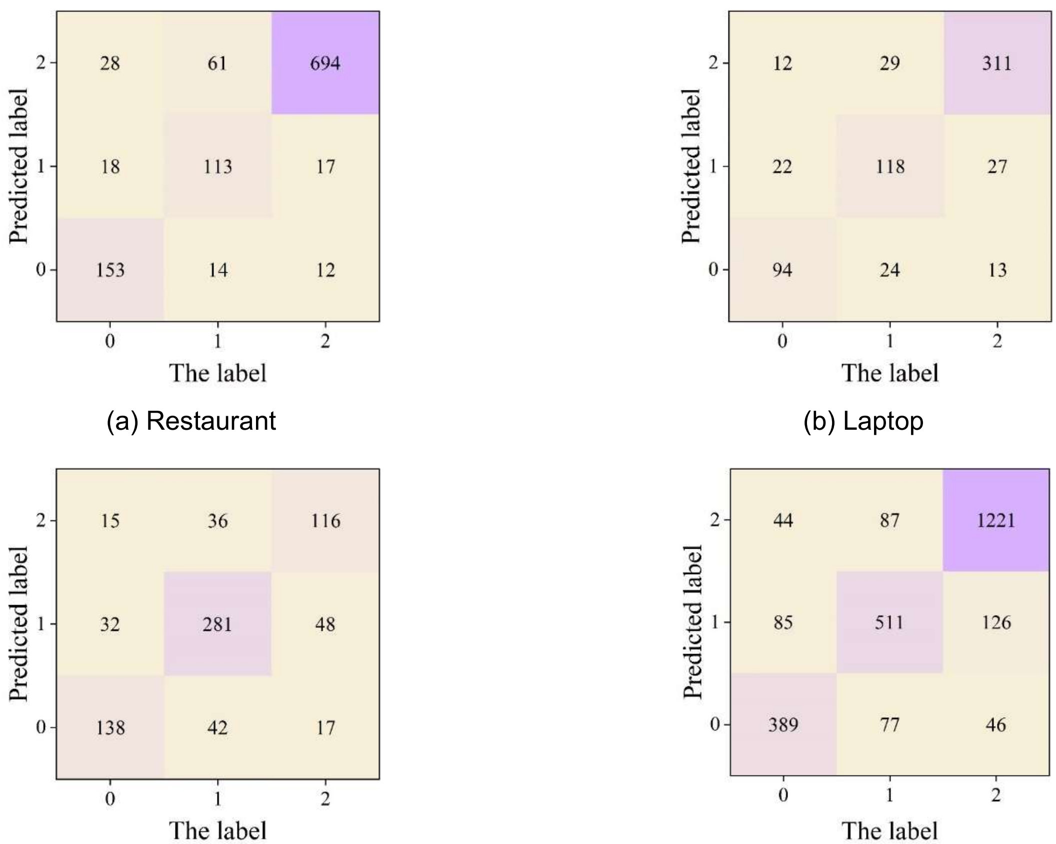
III. C. 1) Emotion classification results

In this paper, we conduct experiments on 2 datasets from Sem Eal-2014v task 4 with Twitter dataset and compare 6 baseline models, the results of the experiments are shown in Table 5. As can be seen from the table, the average F1 value in this experiment is improved more compared to the accuracy, which is improved by 2.79%, 1.81% and 0.45% on the 3 datasets respectively,indicating that the model in this paper is able to check both the accuracy and completeness of the model on multiple sentiment polarities.

Table 5: Accuracy and Macro F1 of models

Models	Restaurant		Laptop		Twitter	
	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1
TNET-LF	0.8064	0.7145	0.7502	0.6983	0.7753	0.7468
CDT	0.8252	0.7444	0.7760	0.7268	0.7921	0.7691
BERT-SPC	0.8468	0.7740	0.794	0.7472	0.781	0.7539
BERT-AEN	0.8334	0.7418	0.8034	0.7600	0.7926	0.7638
TD-GAT	0.8302	-	0.8051	-	-	--
DREGCN	0.8383	0.7621	0.7978	0.7606	-	--
GNNRL	0.8656	0.8019	0.8113	0.7787	0.7984	0.7736

The confusion matrix of the classification results of the model in this paper is shown in Figure 4. The confusion matrix in Fig. 4 visualizes the classification of the model on different categories, and the four sub-figures show the confusion matrix of the model on Res-taurant, Laptop, Twitter and overall data, respectively. Where 0, 1, and 2 denote the sentiment polarity of negative, neutral, and positive, respectively. Figure 4(d) overall data confusion matrix shows that out of the 1221 samples predicted as positive by the model (column 3), 126 data samples are actually neutral samples and only 46 data samples are actually negative samples, which indicates that this paper's model has a better classification performance on the sentiment polarity of positive. The model in this paper has the best classification performance on the Restaurant dataset, which is more in line with the "proximity principle".



(a) Restaurant (b) Laptop

(c) Twitter (d) All

Figure 4: Confusion matrix

III. C. 2) Cross-Language Sentiment Prediction

The cross-language sentiment classification prediction results with English as the source language and Chinese, French, German, Japanese, Korean and Thai as the target languages are shown in Table 6, with AR denoting the accuracy rate. In the table, the optimal prediction accuracy and F1 values for each target language are bolded, and the suboptimal values are underlined. In the experiments of Table 6, the proposed model sets the dimensionality of word vectors to 50 dimensions, α takes the value of 0.8, and adopts GNNRL and CNN as the linguistic generators for joint feature extraction in source and target languages, respectively.

The experimental results show that the proposed model outperforms the machine translation-based, Bi_random and Bi_W2V methods for cross-language sentiment classification on six different languages, namely Chinese, French, German, Japanese, Korean and Thai, which verifies the effectiveness of the graph neural network-based approach for sentiment analysis of text representations.The Upper method provides the cross-language predictive categorization performance of the cross-language sentiment predictions that can be achieved by the model. Upper value. It can be seen that the proposed model GNNRL has an accuracy and F1 value of 0.839 and 0.867 for German, which are close to the Upper method's 0.881 and 0.883, respectively.

CLCDSA gets the best sentiment prediction accuracy on French and Japanese with 0.815 and 0.710, respectively, which is higher than the CNN model's 0.765 and 0.695. However, the generalization performance of the proposed model is more prominent across languages and datasets when looking at six different target languages. On the same Amazon dataset, the prediction accuracy and F1 value of the GNNRL model are better than CLCDSA on German and Chinese; when cross-language across datasets, i.e., using the English dataset of Amazon user reviews as the data source, predicting the movie review dataset with the target language of Korean and the product dataset with the target language of Thai, the GNNRL has an obvious advantage over CLCDSA.

Table 6: Cross-language emotional prediction

Method	French		German		Japanese		Chinese		Korean		Thai	
	AR	F1	AR	F1	AR	F1	AR	F1	AR	F1	AR	F1
Upper method	0.891	0.892	0.881	0.883	0.813	0.801	0.822	0.745	0.767	0.761	0.873	0.782
Machine translation	0.661	0.714	0.753	0.746	0.676	0.701	0.661	0.714	0.655	0.702	0.640	0.669
Bi_random	0.647	0.815	0.587	0.741	0.610	0.609	0.605	0.7	0.551	0.694	0.619	0.683
Bi_W2V	0.732	0.816	0.757	0.741	0.599	0.636	0.653	0.731	0.548	0.738	0.615	0.656
GNNRL	0.759	0.869	0.839	0.867	0.679	0.676	0.681	0.769	0.646	0.756	0.674	0.713
CNN	0.765	0.873	0.845	0.853	0.695	0.746	0.742	0.771	0.637	0.728	0.654	0.670
CLCDSA	0.815	0.820	0.809	0.816	0.710	0.740	0.643	0.687	0.641	0.706	0.626	0.659

III. C. 3) Visual analysis of word-vector representations of words

In order to analyze how the graph neural network-based text representation method can better take into account the word semantic and sentiment feature information compared to Word2Vec from a linguistic and semantic point of view, in this section, a visualization method is used to compare the word vector representations obtained from the GNNRL model and the Word2Vec model. The principal component analysis (PCA) method is used to downscale the word vector representations obtained in the experiments and finally output them in the 2D plane.

Figures 5 and 6 show the output of the two sets of words visualized in the 2D plane under Word2Vec and GNNRL representations, respectively. In order to be able to clearly see the results of the visualized representations, a small number of words were selected as examples for the experiment. Each point in the figure represents the 2D plane embedding result of the high-dimensional word vectors of a word after PCA dimensionality reduction, and the closer the word vector representations of the two points are, the closer they are in the 2D plane.The results of the word vector representations of Word2Vec are on the left side of the figure; the results of the GNNRL representations are on the right side of the figure.

Figure 5 shows a visualization of a group of words "good", "delicious", "bad", "exciting", "happy", "beautiful" in a two-dimensional plane. The emotional polarity of this group of words is obvious, and it can be seen that the text representation of GNNRL takes into account the emotional feature information of the words, and can distinguish words with different emotional polarities.

On the basis of Figure 5, a few words with close semantic meanings: "dog", "cat", and "bird" are added, and a few words are randomly removed, and the visualization results are shown in Figure 6. It can be seen that the Word2vec model has more advantages in semantic representation, and is able to cluster the semantically similar words "dog", "cat", and "bird", but the words "hate" and "exciting" still overlap. However, the textual representation of GNNRL can still clearly distinguish the emotional polarity of words, and "hate" is a negative word with an emotional polarity, which has obvious semantic distance from other words.

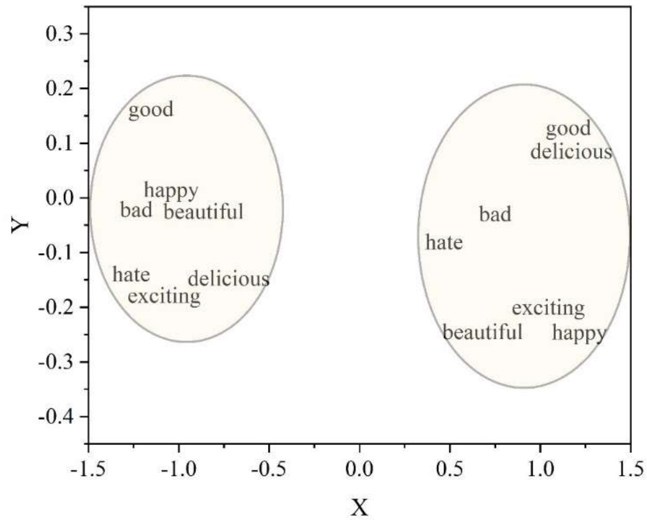


Figure 5: The model text represents the example 1

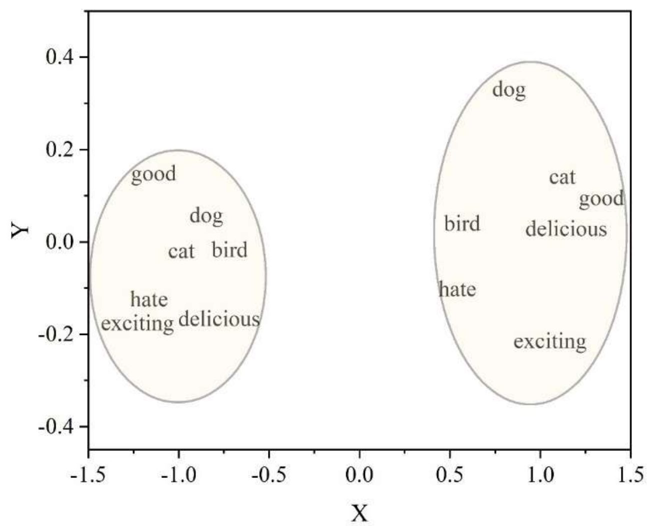


Figure 6: The model text represents the example 2

III. C. 4) Validation of model generalization

In order to verify the generality of GNNRL on aspect-level sentiment analysis tasks, this paper expands the data samples, tests them on the REST15 and REST16 datasets, and analyzes them in comparison with the baseline model TNET-LF and the model GNNRL* that is trained only on Restaurant, and the experimental results are shown in Table 7. The experimental results show that GNNRL still has high classification accuracy and average F1 value on the new sample set; the directly migrated model GNNRL* also has not low classification performance in the untrained case, which proves that GNNRL can effectively extract the features of the text, accurately discriminate the sentiment polarity of the given aspectual words, and has a certain degree of versatility in the task of aspect-level sentiment analysis.

Table 7: Experimental results of different models

Model	REST15		REST16	
	Accuracy	Macro F1	Accuracy	Macro F1
TNET-LF	0.7928	0.6051	0.9012	0.7147
GNNRL*	0.7814	0.6078	0.8739	0.6349
GNNRL	0.8438	0.7371	0.9223	0.7561

IV. Conclusion

This paper proposes a text sentiment analysis model based on graph neural networks and representation learning, which utilizes graph convolutional neural networks for encoding, screening features, avoiding the problem of partial feature loss thus improving the feature extraction capability of the model. In order to verify the effectiveness of the model relevant datasets are selected for testing. In cross-language document representation, the GNNRL model is able to improve the F1 value of cross-language text sentiment classification by 12.35% to 37.83% on the basis of the bilingual word vector method. Taking English labeled text data as the source language, the sentiment polarity prediction is performed on unlabeled texts in Chinese, French, German, Japanese, Korean and Thai, respectively, and the GNNRL model performs better than other sentiment prediction methods in all six languages.

References

[1] Alshamsi, A., Bayari, R., & Salloum, S. (2020). Sentiment analysis in English texts. Advances in Science, Technology and Engineering Systems Journal, 5(6).

[2] Gamallo, P., & Garcia, M. (2014, August). Citius: A naive-bayes strategy for sentiment analysis on english tweets. In Proceedings of the 8th international Workshop on Semantic Evaluation (SemEval 2014) (pp. 171-175).

[3] Wang, H. (2022). Online Corpus Construction of English Text Collection, Data Cleaning, and Similarity Analysis. Mobile Information Systems, 2022(1), 3105790.

[4] Rao, L. (2022). Sentiment analysis of english text with multilevel features. Scientific Programming, 2022(1), 7605125.

[5] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), 4-24.

[6] Veličković, P. (2023). Everything is connected: Graph neural networks. Current Opinion in Structural Biology, 79, 102538.

[7] Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. Ieee Access, 8, 193907-193934.

[8] Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2018). Network representation learning: A survey. IEEE transactions on Big Data, 6(1), 3-28.

[9] Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., & Bengio, Y. (2021). Toward causal representation learning. Proceedings of the IEEE, 109(5), 612-634.

[10] Guo, W., Wang, J., & Wang, S. (2019). Deep multimodal representation learning: A survey. Ieee Access, 7, 63373-63394.

[11] Li, Y., Yang, M., & Zhang, Z. (2018). A survey of multi-view representation learning. IEEE transactions on knowledge and data engineering, 31(10), 1863-1883.

[12] Chen, F., Wang, Y. C., Wang, B., & Kuo, C. C. J. (2020). Graph representation learning: a survey. APSIPA Transactions on Signal and Information Processing, 9, e15.

[13] Liu, Z., Kitouni, O., Nolte, N. S., Michaud, E., Tegmark, M., & Williams, M. (2022). Towards understanding grokking: An effective theory of representation learning. Advances in Neural Information Processing Systems, 35, 34651-34663.

[14] Radulescu, A., Shin, Y. S., & Niv, Y. (2021). Human representation learning. Annual Review of Neuroscience, 44(1), 253-273.

[15] Khoshraftar, S., & An, A. (2024). A survey on graph representation learning methods. ACM Transactions on Intelligent Systems and Technology, 15(1), 1-55.

[16] Thakoor, S., Tallec, C., Azar, M. G., Munos, R., Veličković, P., & Valko, M. (2021, May). Bootstrapped representation learning on graphs. In ICLR 2021 workshop on geometrical and topological representation learning.

[17] Donahue, J., & Simonyan, K. (2019). Large scale adversarial representation learning. Advances in neural information processing systems, 32.

[18] Deng, X., Sun, H., Lees, A., Wu, Y., & Yu, C. (2022). Turl: Table understanding through representation learning. ACM SIGMOD Record, 51(1), 33-40.

[19] Ericsson, L., Gouk, H., Loy, C. C., & Hospedales, T. M. (2022). Self-supervised representation learning: Introduction, advances, and challenges. IEEE Signal Processing Magazine, 39(3), 42-62.

[20] Zhang, Y., Shen, D., Wang, G., Gan, Z., Henao, R., & Carin, L. (2017). Deconvolutional paragraph representation learning. Advances in Neural Information Processing Systems, 30.

[21] Ju, W., Fang, Z., Gu, Y., Liu, Z., Long, Q., Qiao, Z., ... & Zhang, M. (2024). A comprehensive survey on deep graph representation learning. Neural Networks, 106207.

[22] Dehmamy, N., Barabási, A. L., & Yu, R. (2019). Understanding the representation power of graph neural networks in learning graph topology. Advances in Neural Information Processing Systems, 32.

[23] Waikhom, L., & Patgiri, R. (2023). A survey of graph neural networks in various learning paradigms: methods, applications, and challenges. Artificial Intelligence Review, 56(7), 6295-6364.

[24] Wang, Y., Wang, J., Cao, Z., & Barati Farimani, A. (2022). Molecular contrastive learning of representations via graph neural networks. Nature Machine Intelligence, 4(3), 279-287.

[25] Chengpei Xu,Wenhao Zhou,Zhixiong Huang,Yuanfang Zhang,Yan Zhang,Weimin Wang & Feng Xia. (2025). Fusion-based graph neural networks for synergistic underwater image enhancement. Information Fusion,117,102857-102857.

[26] Zuo Yuanyuan,Che Lixuan & Zhang Liping. (2023). The development of robotics courses for young children under vector space model. . PloS one,18(10),e0293397-e0293397.