DOI: 10.19884/j.1672-5220.202503006

Integrating Hierarchical Dependencies and Correlation for Multi-Label Text Classification

ZHU Jiaxin, LIU Guohua*
School of Information and Intelligent Science, Donghua University, Shanghai 201620, China

Abstract: This study addresses the challenges in scientific paper classification caused by the exponential growth of scientific papers, which intensifies demands for efficient manuscript categorization and cross-disciplinary knowledge retrieval. To overcome limitations of conventional methods in handling multi-label scenarios and domain-specific nuances, a novel multi-label text classification model is proposed, integrating semantic-aware label correlation modeling with advanced text representation learning. This model constructs a discipline-aware label graph through a graph attention network, incorporating hierarchical information and co-occurrence patterns from domain knowledge, while employing DeBERTaV3 to capture deep contextual semantics through its enhanced Transformer architecture. Text embeddings and label representations undergo systematic fusion, with hybrid sampling strategies and text augmentation techniques mitigating data imbalance issues. Experimental results demonstrate that the proposed model achieves higher performance in cross-disciplinary text classification compared to existing state-of-the-art methods.

Keywords: multi-label text classification; graph attention network; DeBERTaV3; label representation learning
CLC number: TP391.1 **Document code**: A
Article ID: 1672-5220(2026)05-0121-10

0 Introduction

With the vigorous development of global research activities, the number of scientific papers is growing at an astounding rate. This phenomenon not only increases the difficulty for researchers to access and filter information, but also brings new difficulties to the editorial departments of journals and magazines. Specifically, journal editors need to quickly and accurately categorize a large volume of submitted manuscripts in order to assign them to suitable reviewers and ensure that readers can easily find articles related to specific topics and fields. To meet these challenges, it is essential to adopt advanced technological methods.

Text classification originated in the 1960s but became a major branch of the information field in the 1990s[1]. The methods of text classification can be divided into machine learning-based methods and deep learning-based methods. Machine learning-based text classification involves learning from pre-classified documents to obtain label features and build an automatic classifier. Compared with traditional manual classification, it processes information more quickly and possesses strong domain portability.

1) Machine learning-based text classification

Machine learning-based text classification requires feature engineering as a first step, where text features are extracted through preprocessing and then fed into a classifier. Classic algorithms include the *k*-nearest neighbor (KNN)[2], decision trees[3], and support vector machines (SVMs)[4]. Multi-label text classification is more complex than single-label classification because it involves identifying multiple labels corresponding to a text. The main solutions to this issue include problem transformation and algorithm adaptation. Problem transformation methods convert multi-label issues into multiple binary classification problems, with typical algorithms being binary relevance (BR)[5], label powerset (LP)[6], and classifier chain (CC)[7]. Algorithm adaptation methods directly handle multi-label classification, with typical algorithms including multi-label decision trees (ML-Decision Tree)[8], ranking support vector machine (Ranking-SVM)[9], and multi-label *k*-nearest neighbor (ML-KNN)[10].

To address the issue of label imbalance, there are data-level oversampling strategies, such as the interpolation method proposed by Chawla et al.[11]. Additionally, enhancing tail-label sample learning through improved loss functions is also possible, as demonstrated by the improved balanced loss function method proposed by Huang et al.[12].

2) Deep learning-based text classification

In recent years, deep learning has made significant progress in multi-label text classification. Early models such as Word2Vec[13] and GloVe[14] laid the foundation for text representation by training word vectors that contain semantic and syntactic information. With the advent of new-generation pre-trained language models like ELMo[15] and BERT[16], deep learning has made

Received date: 2025-03-06
* Correspondence should be addressed to LIU Guohua, email: ghliu@dhu.edu.cn
Citation: Zhu J X, Liu G H. Integrating hierarchical dependencies and correlation for multi-label text classification [J]. *Journal of Donghua University (English Edition)*, 2026, 43(5): 121-130.

substantial breakthroughs in text semantic understanding. These models, through deeper neural network structures, successfully address polysemy, making text classification more accurate. Models based on recurrent neural networks (RNNs)[17], convolutional neural networks (CNNs)[18], and graph neural networks (GNNs)[19] show outstanding performance in multi-label text classification tasks. They effectively extract text features and relationships between labels, significantly enhancing classification results.

In 2014, Kim et al.[20] proposed a text classification model based on CNNs (TextCNN), which achieved notable results at the time by training word vectors using the Word2Vec model and utilizing a layer of multi-scale convolution. Besides RNNs and CNNs, some studies have proposed to use GNNs to model graph-structured relations, such as word co-occurrence within a document and heterogeneous links among words, documents, and labels, thereby improving text representation learning. Among all types of GNNs, graph convolutional networks (GCNs)[21] and their variants are the most popular because they efficiently and conveniently combine with other neural networks, achieving excellent results in many applications. Essentially, GCNs perform convolutional operations by leveraging relationships among neighboring nodes in the graph, using dependency syntax trees or word co-occurrence information to better capture internal text information. Graph attention networks (GATs)[22] introduce masked self-attention layers, enabling nodes to adaptively assign different weights to their neighboring features without expensive matrix operations or prior knowledge of the graph structure.

Additionally, the information contained in the labels themselves plays a significant role in multi-label text classification tasks. Through self-attention and label attention mechanisms, models can better integrate text and label features, enhancing classification performance. Xiao et al.[23] proposed the label-specific attention network (LSAN) and a Transformer-based multi-task learning model, both of which demonstrated good results in obtaining joint embeddings of text and labels and enhancing label prediction outcomes. Vu et al.[24] utilized GCNs to model label correlations by designing a data-driven label correlation matrix and combining it with fine-grained document information extracted by RoBERTa for classification. However, generating the correlation matrix based solely on the co-occurrence frequency of labels lacks robustness. There remain substantial research gaps and application potential in developing more effective models for multi-label text classification tasks.

Currently, most existing text classification methods are developed for tasks such as news classification, sentiment analysis, and patent classification. Text in these domains typically has relatively clear and singular topic labels. However, scientific papers exhibit unique characteristics that make the direct application of these methods challenging.

Compared with news reports, scientific papers generally exhibit more complex logical structures, including sections such as the introduction, methodology, results, and discussion, each containing distinct information. They often employ highly specialized language and terminology, presenting diverse and complex content with significant differences between fields. Modern research is increasingly interdisciplinary, and a single paper may cover multiple research areas, making it difficult for a single label to fully encapsulate its content.

These characteristics reveal significant limitations in existing methods. Such methods may struggle to capture the subtle professional distinctions or cross-disciplinary content, and they may not accommodate the common scenario in scientific papers where a document belongs to multiple categories simultaneously. Thus, there is a need for a more refined and multi-dimensional multi-label classification method.

Given these issues, a novel multi-label text classification model is proposed to fully leverage hierarchical information and co-occurrence patterns of labels based on understanding label semantics. The proposed model divides the multi-label text classification task into three parts. First, it constructs a label relationship graph via a GAT, utilizing prior knowledge of disciplinary hierarchies and label co-occurrence patterns. Second, it employs DeBERTaV3 to extract deep contextual semantic features from the input text. These features are subsequently integrated with the label representations derived from the GAT to produce the final classification results. Finally, a classifier is trained to predict the probability of each label corresponding to the text.

1 Proposed Model

1.1 Problem statement

The problem addressed in this study is to assign classification labels to a given scientific paper. The key objective is to learn a mapping function from the text to a set of labels, which can be formulated as follows.

Let set $P = \{p_1, p_2, \cdots, p_n\}$ represent a collection containing n scientific papers, where p_n represents the nth paper. The label space $Y = \{c_1, c_2, \cdots, c_k\}$ comprises k category labels. The feature space $\boldsymbol{X} = \mathbf{R}^{n \times d}$ corresponds to the d-dimensional feature vectors for n documents. P' and Y' are the training sets constructed by extracting papers and the corresponding labels for each paper from the dataset. The task is to learn a mapping function $G(\cdot) = \boldsymbol{X} \rightarrow 2^Y$ from the training set $P' = \{(\boldsymbol{x}_i, Y') \mid 1 \leqslant i \leqslant n, Y' \subset Y\}$, where $\boldsymbol{x}_i \in \boldsymbol{X}$ is the feature vector corresponding to p_i, and $Y' \in 2^Y$ is the set of labels for $\boldsymbol{x}_i$.

In order to learn the mapping function $G(\cdot)$, the feature vector $\boldsymbol{x}_i$ is combined with its label information Y'. The label information is learned using a graph attention network, while the text embeddings are obtained using

DeBERTaV3.

1.2 Text representation learning

The text embedding is the process of converting words or sentences in a text into low-dimensional vector representations, ensuring that semantically similar texts are closer in the embedding space. This facilitates the model's learning, understanding, and processing of natural language. DeBERTa[25] introduces two novel components to improve BERT. First, it adopts the disentangled attention (DA) mechanism, which uses two separate vectors to record content information and positional information, respectively. When calculating the attention weights between words, it computes them through separate content and relative position matrices. Additionally, to compensate for the lack of absolute positional information in DA, DeBERTa adds absolute positional information to the masked language model (MLM) decoding layer to enhance the MLM.

However, the MLM task focuses on predicting masked words and does not predict every token, resulting in inefficient utilization of samples and requiring substantial computational resources for pretraining. Clark et al.[26] proposed a pretraining method to discriminate whether a particular token in a corrupted input was original or replaced by a generator, thereby fully utilizing every token in the training data. Hadsell et al.[27] discovered that models trained with the ELECTRA method exhibit the tug-of-war dynamics. He et al.[28] suggested that this occurred because the discriminator and generator in ELECTRA shared the same embeddings, but their training objectives were fundamentally different, leading to a certain degree of conflict. Experiments using separate embeddings revealed worse performance on downstream tasks. To address this, a gradient separation embedding sharing method was proposed, allowing the discriminator and generator to share embeddings while preventing gradients of the discriminator from propagating to the generator's embeddings. This approach avoids the tug-of-war effect while retaining the advantages of shared embeddings. Building upon these improvements, DeBERTa was further enhanced to develop DeBERTaV3.

In this study, DeBERTaV3 is primarily used to efficiently extract and encode the deep semantic information of texts, generating high-quality text embeddings that provide a solid foundation for subsequent prediction tasks. Through large-scale pretraining, DeBERTaV3 acquires rich linguistic knowledge and semantic information, enabling rapid adaptation to new tasks and offering high-quality initial embedding representations. This reduces the time and resource consumption associated with training from scratch. Its Transformer architecture can capture long-distance dependencies, enhancing the model's understanding of complex sentence structures and contextual information, which is particularly beneficial for multi-label text classification tasks. The multi-layer Transformer encoder progressively extracts and abstracts high-level features from the text, encompassing deep lexical, syntactic, and semantic information, allowing the model to more accurately capture key text features. Additionally, the pretrained DeBERTaV3 model provides strong contextual representations and can be adapted to the downstream classification task by training the subsequent projection and classification layers while keeping the encoder parameters frozen. High-quality text embeddings also accelerate the convergence speed of the model, reduce the risk of overfitting, and facilitate effective learning of complex relationships between subject categories, thereby enhancing classification accuracy and reliability.

1.3 Label representation learning

1.3.1 Label graph attention network

GCNs are a type of neural network models suitable for node classification in graph structures. In a single GCN layer, each node aggregates information from its directly connected neighbors to update its own features. By stacking multiple GCN layers, each node can aggregate information from multi-hop neighboring nodes. However, information propagation in GCNs is relatively slow, and the utilization of a node's degree information in updating node features is insufficient. GATs enhance GCNs by incorporating attention mechanisms. The core idea is to enable nodes to focus on their neighbors through a self-attention mechanism, which learns different weights for each neighbor of a node. This allows the model to differentiate the importance of information from various neighboring nodes to the current node. Additionally, multi-head attention is employed to expand the model's capacity and stabilize the learning process. For a single GAT layer, the input is a set of node features: $z = \{z_1, z_2, \cdots, z_i, \cdots, z_Q\}$, $z_i \in \mathbf{R}^F$, where Q is the number of nodes in the graph, and F is the number of features per node. The output of each layer is an updated set of node features: $z' = \{z'_1, z'_2, \cdots, z'_i, \cdots, z'_Q\}$, $z'_i \in \mathbf{R}^{F'}$. During this process, each node feature undergoes a linear transformation using a weight matrix $\boldsymbol{W} \in \mathbf{R}^{F' \times F}$. Subsequently, a self-attention mechanism (a single-layer feedforward neural network, parametrized by a weight vector $\boldsymbol{a} \in \mathbf{R}^{2F'}$) $\boldsymbol{a}: \mathbf{R}^{F'} \times \mathbf{R}^{F'} \rightarrow \mathbf{R}$ is applied at each node to compute an attention coefficient $e_{ij} = \boldsymbol{a}(\boldsymbol{W}z_i, \boldsymbol{W}z_j)$, which represents the importance of node j to node i. N_i is the neighboring node set of node i, and $j \in N_i$ is one of the neighboring nodes. A leaky rectified linear function (LeakyReLU) is used, and a multi-head mechanism is adopted. In hidden GAT layers, the outputs of multiple attention heads are concatenated to enrich the representation:

$$\alpha_{ij} = \frac{\exp(\mathrm{LeakyReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}z_i \| \boldsymbol{W}z_j]))}{\sum_{q \in N_i} \exp(\mathrm{LeakyReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}z_i \| \boldsymbol{W}z_q]))},$$

where α_{ij} represents the weight value of node j to node i; $\|$ represents the concatenation operation for heads due to multi-head mechanism; q denotes the number of

neighbors of node i. In the final GAT layer, the multi-head outputs are averaged to obtain the final label representation:

$$z'_i = \sigma\left(\frac{1}{H}\sum_{h=1}^{H}\sum_{j\in N_i}\alpha_{ij,h}W_h z_j\right),$$

where H is the number of heads; $\alpha_{ij,h}$ is the normalized attention coefficient computed by the hth attention mechanism (α_h); W_h is the corresponding input linear transformation's weight matrix; σ is a nonlinear activation function.

1.3.2 Correlation matrix

To more accurately capture complex relationships among categories, this study proposes a method for constructing a label graph. This method not only considers the co-occurrence patterns at both the main-category and subcategory levels but also incorporates the hierarchical relationships between subcategories and main categories, as well as the semantic similarities among subcategories. By integrating these factors, a richer and more comprehensive relationship matrix is constructed, thereby enhancing the performance of multi-label text classification tasks. The co-occurrence patterns at both the main-category and subcategory levels reflect the frequency with which different main categories appear simultaneously in the dataset, aiding in the identification of correlations between main categories. The hierarchical relationships reveal the organizational structure of categories, enhancing the model's understanding of how categories are organized. The semantic similarities among subcategories are further enhanced by computing the word vector similarities of their names or descriptions, thereby improving the model's ability to understand subtle differences between categories.

The construction method is outlined in Algorithm 1. $\boldsymbol{R}$ refers to the relationship matrix of a specific aspect in the label set, $\boldsymbol{R}_{co}$ denotes the co-occurrence matrix, $\boldsymbol{R}_{sem}$ represents the semantic similarity matrix, and $\boldsymbol{R}_{hie}$ indicates the hierarchical relationship matrix (where only elements representing main-category and subcategory relationships may have coefficients). Here, M denotes the main-category set, S represents the subcategory set, $\mathbf{R}$ stands for the feature space, and $\varphi = \psi(i, j)$ is a binary function that indicates whether a subcategory belongs to a given main category.

This graph initialization algorithm captures label relationships through three key aspects: co-occurrence patterns (measuring how often labels appear together), semantic similarity (using embedding cosine similarity with a threshold of θ to find conceptually related labels), and hierarchical links (binary connection $\varphi = 1$ for parent-child category pairs). Co-occurrence addresses statistical dependencies in observed data, while semantic similarity helps connect labels with sparse co-occurrence but strong conceptual ties. Hierarchical constraints preserve structural parent-child relationships in the taxonomy. This design intentionally allows overlapping relationships (e.g., semantically similar co-occurring labels) to reinforce connection strengths through combined weights.

Algorithm 1 Constructing label graph and setting initial edge weights

Input: Main-category set $M = \{m_1, m_2, \cdots, m_{main_max}\}$, subcategory set $S = \{s_1, s_2, \cdots, s_{sub_max}\}$, semantic similarity threshold θ, co-occurrence weight β_1, semantic similarity weight β_2, hierarchical weight β_3

Output: Initial correlation matrix $\boldsymbol{A}$

Initialize edge sets: $\boldsymbol{R}_{co} \leftarrow \varnothing$, $\boldsymbol{R}_{sem} \leftarrow \varnothing$, $\boldsymbol{R}_{hie} \leftarrow \varnothing$

Calculate main-category co-occurrence frequencies $C(m_i, m_j)$:
for each main-category pair $(m_i, m_j) \in M \times M$ **do**
  if $C(m_i, m_j) > 0$ **then**
    add edge (m_i, m_j) with weight $C(m_i, m_j)$ to $\boldsymbol{R}_{co}$

Build hierarchical links:
for each $s_p \in S$ **do**
find parent $m_i \in M$, add edge (m_i, s_p) with $\gamma = 1$ to $\boldsymbol{R}_{hie}$

Calculate subcategory co-occurrence frequencies $C(s_p, s_q)$, $p, q \in [1, sub_max)$:
for each pair $(s_p, s_q) \in S \times S$ **do**
  if $C(s_p, s_q) > 0$ **then**
    add edge (s_p, s_q) with weight $C(s_p, s_q)$ to $\boldsymbol{R}_{co}$

Calculate subcategory semantic similarity:
for each pair $(s_p, s_q) \in S \times S$ **do**
$\mathrm{sim}(s_p, s_q) = z_{s_p} \cdot z_{s_q} / |z_{s_p}||z_{s_q}|$
  if $\mathrm{sim}(s_p, s_q) > \theta$ **then**
    add edge (s_p, s_q) with weight $\mathrm{sim}(s_p, s_q)$ to $\boldsymbol{R}_{sem}$

Merge all edge sets: $\boldsymbol{R} = \boldsymbol{R}_{co} \cup \boldsymbol{R}_{hie} \cup \boldsymbol{R}_{sem}$

Initialize correlation matrix $\boldsymbol{A} \in \mathbf{R}^{|L|\times|L|}$ as a zero matrix, where $L = M \cup S$

Normalize co-occurrence edges, here g_i is the degree of node i:
if $i \neq j$, $\mathrm{Norm}(i, j) = C(i, j)/\sqrt{g_i g_j}$
if $i = j$, $\mathrm{Norm}(i, j) = 1/g_i$

Populate matrix $\boldsymbol{A}$:
for each edge $(i, j) \in \boldsymbol{R}$ **do**
weight_sum = 0
  if $(i, j) \in \boldsymbol{R}_{co}$ **then**
    weight_sum += $\beta_1 \cdot \ln(1 + \mathrm{Norm}(i, j))$
  if $(i, j) \in \boldsymbol{R}_{sem}$ **then**
    weight_sum += $\beta_2 \cdot \mathrm{sim}(i, j)$
  if $(i, j) \in \boldsymbol{R}_{hie}$ **then**
    weight_sum += $\beta_3 \cdot \psi(i, j)$
A_{ij} = weight_sum
Return $\boldsymbol{A}$

1.4 Multi-label text classification model

This study proposes a model for multi-label classification of scientific papers, which is based on

DeBERTaV3 and incorporates a label representation learning component. As illustrated in Fig. 1, the overall framework is divided into two main components: label representation learning and text representation learning.

In the label representation learning component, the input labels are first transformed into preliminary label embeddings through ξ_{emb}. These label embeddings are then iteratively updated through GAT layers to capture the mutual influences and correlations among the labels. The initialization parameters for the first GAT layer are derived from Algorithm 1. After being input into the network, attention calculations are performed. Through iterative processing by the GAT layers, the label embeddings $\boldsymbol{z}_1$ to $\boldsymbol{z}_k$ are aggregated into a unified label embedding $\boldsymbol{Z}$:

$$\boldsymbol{Z} = f_{\text{gat}}(\boldsymbol{A},\ \boldsymbol{Y},\ \boldsymbol{W}_{\text{gat}}),$$

where $\boldsymbol{W}_{\text{gat}}$ is the weight matrix of the GAT layers' function f_{gat}; $\boldsymbol{Y}$ is the set of labels' embeddings ($\boldsymbol{Y} \in \mathbf{R}^{k\times d}$, k indicates the number of labels, and d is the dimension of embeddings, and after GAT layers will be shaped in d'). All the label embeddings are concatenated to form a label representation matrix.

In the text representation learning component, each input text p_i undergoes tokenization and turns into a series of text tokens $\{\text{tok}_1, \text{tok}_2, \cdots, \text{tok}_m\}$. Each token is fed into the DeBERTaV3 model for encoding with a frozen parameter of η. DeBERTaV3 utilizes its internal Transformer architecture to perform deep learning on these tokens, generating corresponding embedding representations $\{\boldsymbol{E}_1, \boldsymbol{E}_2, \cdots, \boldsymbol{E}_m\}$. These embeddings encapsulate the semantic information of the text, capturing the complex relationships between words and contextual dependencies, and are concatenated to form a comprehensive text embedding $\boldsymbol{E}$. This concatenated embedding is then mapped to a low-dimensional space through a linear layer ρ, resulting in the final text embedding representation $\boldsymbol{T}_{p_i}$:

$$\boldsymbol{T}_{p_i} = \rho(\delta_{\text{DeBERTaV3}}(p_i,\ \eta)).$$

Finally, the cosine similarity between the label embeddings and the text embeddings of p_i is computed to obtain the predicted scores for each label as $\hat{\boldsymbol{y}} = \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{T}_{p_i}$.

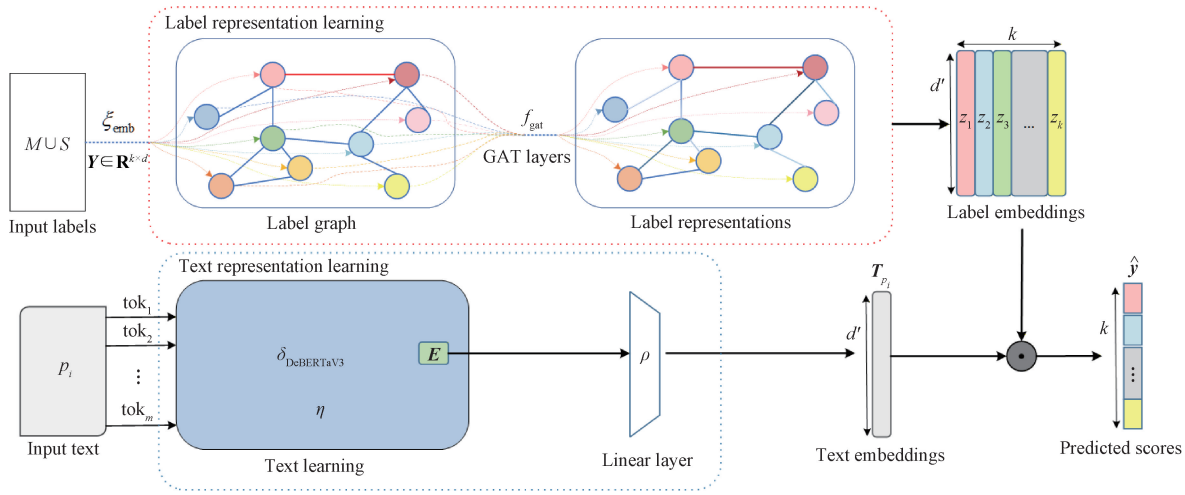


Fig. 1 Architecture of multi-label text classification model

2 Experiments and Results

2.1 Datasets

In the experiments, two widely recognized benchmark datasets for multi-label text classification were employed to evaluate the proposed model: Web of Science (WOS) with hierarchical multi-label text classification[29] and arXiv Academic Paper Dataset (AAPD)[30]. These datasets consist of documents, and corresponding classification labels of scientific papers collected from their own databases. Specifically designed for multi-label text classification tasks, they both feature rich themes and a hierarchical structure, making them highly suitable for evaluating the proposed model. The statistical information of the datasets is presented in Table 1, where M_{c} is the number of main categories, S_{c} is the number of subcategories, and P_{d} is the number of documents.

Table 1 Statistical information of datasets

Dataset	M_{c}	S_{c}	P_{d}
WOS	7	142	30 855
AAPD	11	146	55 000

To mitigate bias caused by the imbalance of disciplinary categories in the datasets, under-sampling and over-sampling were applied to each category of samples. For the majority categories, random sampling was conducted, while for the minority categories, random over-sampling was performed, including data augmentation techniques such as text duplication and synonym replacement. These approaches were employed

to reconstruct the training set and prevent an overly skewed category distribution.

2.2 Baselines

To demonstrate the effectiveness of the proposed model, this study compares it against several state-of-the-art multi-label text classification models.

1) BERT[16] revolutionizes natural language processing by introducing a Transformer-encoder framework trained on vast text resources. Its profound understanding of linguistic nuances, achieved through bidirectional context processing, has made it an indispensable baseline for developing modern language understanding systems.

2) GCN-MTC[24] is a model based on a single-layer GCN and combines it with fine-grained text representations from a RoBERTa-base model. It incorporates normalization and edge-filtering mechanisms to refine label interactions during information propagation.

3) AttentionXML[31] is a deep learning model based on a multi-label attention mechanism that captures the most relevant parts of the text for each label. It also employs a hierarchical tree structure that allows for the processing of millions of labels.

4) HTTN[32] addresses the issue of insufficient training data in tail label classification by transferring meta-knowledge from data-rich head labels to data-scarce tail labels.

5) LDGN[33] models label-specific text representations and label correlations through a dual graph neural network, thereby capturing complex dependencies for multi-label text classification.

2.3 Settings and evaluation metrics

The proposed multi-label text classification model for scientific papers in this study was trained on a single V100 32 GB GPU. The architecture incorporated a two-layer GAT with an embedding dimension of 512. GloVe was utilized for label representations, while document embeddings were generated using the DeBERTaV3-base model. For the correlation matrix configuration, optimal performance was achieved when parameters were set as shown in Table 2. The training process employed the AdamW optimizer, with a batch size of 32, learning rates of 3×10^{-5} and 5×10^{-5} for WOS and AAPD, trained for 10 and 8 epochs, respectively. Since WOS contained no instance of main-category co-occurrence, β_1 was set as 0.

Table 2 Parameter settings for datasets

Dataset	β_1	β_2	β_3	θ
WOS	0	0.5	0.8	0.6
AAPD	0.6	0.4	0.3	0.6

When evaluating the performance of multi-label text classification models, it is crucial to employ appropriate evaluation metrics. Following Refs. [24, 32-33], this study utilized two evaluation metrics to comprehensively assess the model's performance across various aspects: precision at top t (P@t) and normalized discounted cumulative gain at top t (nDCG@t).

P@t is a metric to evaluate the accuracy of the top t predicted labels. In multi-label text classification tasks, for each input instance, all possible labels are ranked based on their predicted probabilities by the model, and the top t labels are selected as the prediction results for that instance. P@t measures the consistency between these predicted labels and the actual relevant labels:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Y'_{i,t}|}{t},$$

where N denotes the number of samples included in the evaluation; Y_i represents the set of true labels for the ith sample; $Y'_{i,t}$ is the set of the top t labels predicted by the model for sample i based on probability ranking; $|Y_i \cap Y'_{i,t}|/t$ represents the precision of the predictions within the top t labels.

nDCG@t evaluates the retrieval quality by considering the ranking order, and it is particularly suitable for data contexts with hierarchical relationships and distinctions in importance. In the context of multi-label text classification problems, DCG@t can be utilized to assess the ranking of correct labels within the output list. Initially, DCG@t is

$$\sum_{j=1}^{t}\frac{2^{v_{ij}}-1}{\log_2(j+1)},$$

where v_{ij} is a binary value indicating the relevance score of the ith sample at the jth position, that is, it is 1 if the jth predicted label belongs to the set of true labels, and 0 otherwise. Subsequently, the maximum possible DCG value under ideal conditions, known as IDCG, is obtained, which represents the DCG value after optimal sorting according to true relevance. Finally, DCG@t is normalized by using

$$\text{nDCG@}t = \frac{\text{DCG@}t}{\text{IDCG@}t}.$$

A high nDCG@t value signifies that the model not only identifies the relevant labels for document classification but also ranks the most significant and representative labels in the foremost positions.

2.4 Experiment results

In this experiment, both datasets come with predefined training and test splits. The predefined training and test splits of the original datasets were retained for fair comparison. The sampling and augmentation strategies were applied only to the training set, while the test set remained unchanged. Tables 3 and 4 show the comparison between the proposed model and the baseline models in terms of P@t and nDCG@t on the two datasets. Since each sample only has two correct labels ("domain" and "area") on WOS, the evaluation metrics will be P@1 and P@2.

The results in Tables 3 and 4 show that the proposed model slightly but consistently outperforms the baseline models on both datasets. On WOS, the proposed model achieves the best P@1 and P@2 scores, reaching 89.50% and 83.80%, respectively. Compared with GCN-MTC, the strongest baseline model on this dataset, the proposed model improves P@1 by 1.29 percentage points and P@2 by 1.46 percentage points. On AAPD, the proposed model achieves the highest scores on all reported metrics, although the margins over LDGN are relatively small for P@5, nDCG@3, and nDCG@5. This indicates that the proposed model provides consistent but modest improvements on this dataset.

Table 3 Result comparison on WOS

Model	P@1/%	P@2/%
BERT	84.90	78.24
AttentionXML	86.50	80.12
HTTN	85.30	79.45
LDGN	87.80	81.56
GCN-MTC	88.21	82.34
Proposed	89.50	83.80

Table 4 Result comparison on AAPD

Model	P@1/%	P@3/%	P@5/%	nDCG@3/%	nDCG@5/%
BERT	78.37	76.48	67.25	60.78	63.83
AttentionXML	83.04	78.70	69.53	77.86	81.68
HTTN	83.84	79.89	68.77	79.23	82.46
LDGN	86.24	81.95	72.30	83.35	86.85
GCN-MTC	86.70	82.77	71.92	82.89	85.64
Proposed	87.05	83.15	72.36	83.40	86.93

The loss curves of the proposed model on the two datasets are shown in Fig. 2. As training progresses, both the training and validation losses decrease rapidly in the early epochs and then gradually stabilize. The validation loss follows a trend similar to the training loss without an obvious increase in the later stages, indicating that the model converges steadily on both datasets without apparent overfitting.

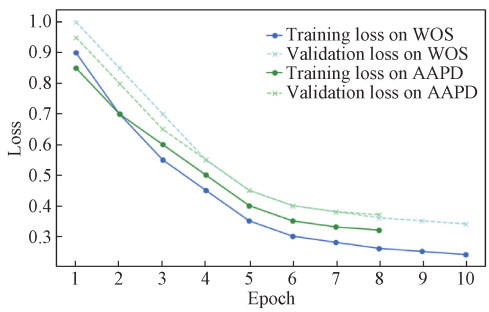


Fig. 2 Loss curves on two datasets

2.5 Ablation studies

This section analyzes the impact of threshold θ on both datasets and examines the effects of key factors on AAPD.

2.5.1 Analysis of semantic similarity threshold

The impact of threshold θ on model performance is shown in Figs. 3 and 4. The P@t metrics exhibit an asymmetric trend of initial rise followed by decline as θ increases from 0.1 to 0.9, peaking at 0.6. At low values, excessive noisy edges from weakly related label pairs degrade graph quality, impairing the GAT's ability to focus on critical relationships, while the performance improves rapidly as θ approaches 0.6. At higher values, the graph becomes overly sparse, losing potentially useful connections and limiting multi-hop semantic propagation, leading to gradual performance degradation. The optimal balance between noise suppression and semantic retention is achieved at $\theta = 0.6$, demonstrating the threshold's critical role in graph structure optimization for text classification tasks.

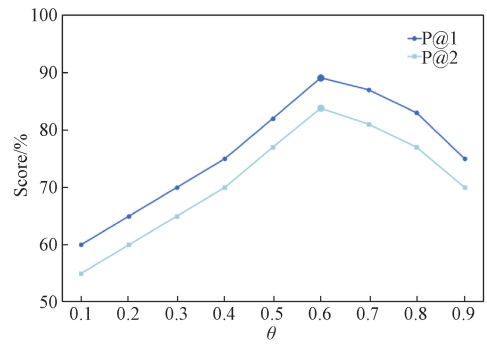


Fig. 3 Impact of θ on metrics on WOS

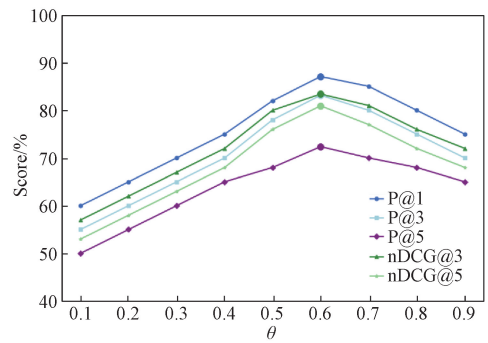


Fig. 4 Impact of θ on metrics on AAPD

2.5.2 Analysis of key factors

To better analyze the impact of three factors on AAPD, ablation experiments were conducted. The ablation combinations and the results are shown in Tables 5 and 6, respectively, where w/o denotes without.

Table 5 Definition of all combinations

Combination	Co-occurrence	Semantic similarity	Hierarchical relation	Parameter setting
Full model	√	√	√	Full setting
w/o Co-occurrence	×	√	√	$\beta_1 = 0$
w/o Semantic	√	×	√	$\beta_2 = 0$
w/o Hierarchy	√	√	×	$\beta_3 = 0$

Table 6 Ablation results and comparison with GCN-MTC

Combination	P@1/%	P@3/%	vs GCN-MTC	
			P@1/%	P@3/%
Full model	87.05	83.15	↑0.35	↑0.38
w/o Co-occurrence	85.90	81.22	↓0.80	↓1.55
w/o Semantic	86.20	82.10	↓0.50	↓0.67
w/o Hierarchy	86.80	82.90	↑0.10	↑0.13

Compared with the GCN-MTC model, the ablation results clearly demonstrate the distinct contributions of the three modules to the overall performance of the proposed model. The full model achieves the best results, with a P@1 value of 87.05% and a P@3 value of 83.15%, outperforming GCN-MTC by 0.35% and 0.38%, respectively. This indicates that the proposed multi-module fusion strategy is effective in improving multi-label prediction performance.

Among the three modules, the co-occurrence frequency module contributes the most to the overall performance. When this module is removed, P@1 decreases from 87.05% to 85.90%, and P@3 decreases from 83.15% to 81.22%. Moreover, the model without the co-occurrence frequency module performs worse than GCN-MTC, with decreases of 0.80% in P@1 and 1.55% in P@3. This result shows that the co-occurrence frequency module plays a fundamental role in capturing label correlations and is especially important for improving the prediction of the top three labels.

The semantic similarity module also has a clear positive effect. After removing this module, P@1 and P@3 decrease to 86.20% and 82.10%, respectively. Compared with GCN-MTC, the model without the semantic similarity module is also slightly inferior, with P@1 and P@3 decreasing by 0.50% and 0.67%, respectively. This suggests that semantic information can effectively supplement co-occurrence statistics by helping the model capture relationships between semantically related labels, thereby improving prediction accuracy.

In contrast, the hierarchical relation module has a relatively limited but still positive contribution, as removing it causes only small decreases in P@1 and P@3. When considering the combined effects of all three modules, the full model achieves a consistent improvement over GCN-MTC, thereby validating the effectiveness of the multi-module fusion strategy. In summary, these three modules, through appropriate weight configurations and their complementary functionalities, not only perform their individual roles effectively but also synergistically optimize the model, resulting in performance that surpasses traditional single-mechanism models.

3 Conclusions

This study presents a multi-label text classification model that integrates a GAT with DeBERTaV3 for deep label representation learning and text representation learning. The label graph construction incorporates hierarchical dependencies, co-occurrence relationships, and semantic similarities between categories. Text embeddings generated by DeBERTaV3 are fused with label features derived from GAT to train the classifier. To address data imbalance, under-sampling and over-sampling strategies combined with text augmentation techniques are applied, balancing sample sizes across categories. Experimental results highlight the effectiveness of combining structured label interactions with robust text semantics, improving accuracy in handling multi-label scenarios with complex category distributions.

Future research could explore the following directions. For model lightweighting, the integration of knowledge distillation and network pruning techniques could be leveraged to develop lightweight inference frameworks suitable for complex scenarios, thereby overcoming computational constraints on edge devices. In self-supervised hierarchical inference, it is essential to construct multi-granular semantic representation spaces, achieving feature disentanglement across different abstraction levels through contrastive learning mechanisms to reduce reliance on manual annotations. Regarding cross-domain adaptation, dynamic meta-learning-based domain adaptation methods could be investigated to enhance model capabilities in heterogeneous scenarios. These advancements are expected to further improve the robustness and scalability of the model in complex scientific paper classification tasks.

References

[1] Dhar A, Mukherjee H, Dash N S, et al. Text categorization: past and present [J]. *Artificial Intelligence Review*, 2021, 54(4): 3007-3054.

[2] Cover T, Hart P. Nearest neighbor pattern

classification [J]. *IEEE Transactions on Information Theory*, 1967, 13(1): 21-27.
[3] Mitchell T M. Machine learning [M]. New York: McGraw-Hill, 1997.
[4] Joachims T. Making large-scale support vector machine learning practical [M]//Advances in Kernel Methods. Cambridge: The MIT Press, 1998.
[5] Boutell M R, Luo J B, Shen X P, et al. Learning multi-label scene classification [J]. *Pattern Recognition*, 2004, 37(9): 1757-1771.
[6] Tsoumakas G, Katakis I. Multi-label classification: an overview [J]. *International Journal of Data Warehousing and Mining*, 2007, 3(3): 1-13.
[7] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification [J]. *Machine Learning*, 2011, 85(3): 333-359.
[8] Clare A, King R D. Knowledge discovery in multi-label phenotype data [C]//Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer, 2001: 42-53.
[9] Elisseeff A, Weston J. A kernel method for multi-labelled classification [M]//Advances in Neural Information Processing Systems 14. Cambridge: The MIT Press, 2002: 681-688.
[10] Zhang M L, Zhou Z H. ML-KNN: a lazy learning approach to multi-label learning [J]. *Pattern Recognition*, 2007, 40(7): 2038-2048.
[11] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique [J]. *Journal of Artificial Intelligence Research*, 2002, 16(1): 321-357.
[12] Huang Y, Giledereli B, Köksal A, et al. Balancing methods for multi-label text classification with long-tailed class distribution [PP/OL]. V2. arXiv (2021-10-15) [2025-05-02]. https://arxiv.org/pdf/2109.04712v2.
[13] Mikolov T, Chen K, Corrado G S, et al. Efficient estimation of word representations in vector space[PP/OL]. V3. arXiv (2013-09-07) [2025-05-02]. https://arxiv.org/pdf/1301.3781.
[14] Pennington J, Socher R, Manning C. GloVe: global vectors for word representation [C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Kerrville: Association for Computational Linguistics, 2014: 1532-1543.
[15] Ilić S, Marrese-Taylor E, Balazs J, et al. Deep contextualized word representations for detecting sarcasm and irony [PP/OL]. V1. arXiv (2018-09-26) [2025-05-02]. https://arxiv.org/pdf/1809.09795.
[16] Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [PP/OL]. V2. arXiv (2019-05-24) [2025-05-02]. https://arxiv.org/pdf/1810.04805.
[17] Elman J L. Finding structure in time [J]. *Cognitive Science*, 1990, 14(2): 179-211.
[18] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [C]//Proceedings of the IEEE. Piscataway: IEEE, 2002: 2278-2324.
[19] Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model [J]. *IEEE Transactions on Neural Networks*, 2009, 20(1): 61-80.
[20] Kim Y. Convolutional neural networks for sentence classification [PP/OL]. V2. arXiv (2014-09-03) [2025-05-02]. https://arxiv.org/pdf/1408.5882.
[21] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [PP/OL]. V4. arXiv (2017-02-22) [2025-05-02]. https://arxiv.org/pdf/1609.02907.
[22] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks [PP/OL]. V3. arXiv (2018-02-04) [2025-05-02]. https://arxiv.org/pdf/1710.10903.
[23] Xiao L, Huang X, Chen B L, et al. Label-specific document representation for multi-label text classification [C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Kerrville: Association for Computational Linguistics, 2019: 466-475.
[24] Vu H, Nguyen M, Nguyen V, et al. Label correlation based graph convolutional network for multi-label text classification [C]//2022 International Joint Conference on Neural Networks (IJCNN). Piscataway: IEEE, 2022: 1-8.
[25] He P C, Liu X D, Gao J F, et al. DeBERTa: decoding-enhanced BERT with disentangled attention [PP/OL]. V6. arXiv (2021-10-06) [2025-05-02]. https://arxiv.org/pdf/2006.03654.
[26] Clark K, Luong M T, Le Q V, et al. ELECTRA: pre-training text encoders as discriminators rather than generators [PP/OL]. arXiv (2020-03-23) [2025-05-02]. https://arxiv.org/pdf/2003.10555.
[27] Hadsell R, Rao D, Rusu A A, et al. Embracing change: continual learning in deep neural networks [J]. *Trends in Cognitive Sciences*, 2020, 24(12): 1028-1040.
[28] He P C, Gao J F, Chen W Z. DeBERTaV3: improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing [PP/OL]. V4. arXiv (2023-03-24) [2025-05-02]. https://arxiv.org/pdf/2111.09543.
[29] Kowsari K, Brown D E, Heidarysafa M, et al. HDLTex: hierarchical deep learning for text

classification [C]//2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). Piscataway: IEEE, 2018: 364-371.
[30] Yang P C, Sun X, Li W, et al. SGM: sequence generation model for multi-label classification [C]//Proceedings of the 27th International Conference on Computational Linguistics. Kerrville: Association for Computational Linguistics, 2018: 3915-3926.
[31] You R, Zhang Z, Wang Z, et al. AttentionXML: label tree-based attention-aware deep model for high performance extreme multi-label text classification[J]. *Advances in Neural Information Processing Systems*, 2019: 5812-5822.
[32] Xiao L, Zhang X L, Jing L P, et al. Does head label help for long-tailed multi-label text classification [J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, 35 (16): 14103-14111.
[33] Ma Q W, Yuan C Y, Zhou W, et al. Label-specific dual graph neural network for multi-label text classification [C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Kerrville: Association for Computational Linguistics, 2021: 3855-3864.

融合层次依赖与关联关系的多标签文本分类

朱嘉欣，刘国华*
东华大学 信息与智能科学学院，上海 201620

摘　要：针对科学论文数量迅速增长所引发的学术文本分类挑战，重点解决了稿件高效分类及跨学科知识检索的难题。针对传统方法在多标签场景处理和领域特异性语义捕捉方面的局限，提出了一种新型多标签文本分类模型：通过集成语义感知的标签关联建模与先进的文本表征学习技术，构建学科感知的标签图。模型通过图注意力网络融合学科层次结构与共现关系等先验知识，同时借助增强型 Transformer 架构的 DeBERTaV3 捕捉深度上下文语义。通过系统化融合文本嵌入与标签表征，结合混合采样策略与文本增强技术有效缓解数据不平衡。实验结果表明，相较于现有方法，本模型在跨学科文本分类任务中展现出更优性能。

关键词：多标签文本分类；图注意力网络；DeBERTaV3；标签表征学习