LabelCoRank: Revolutionizing Long Tail Multi-Label Classification with Co-Occurrence Reranking

YAN YAN, China University of Mining & Technology, Beijing, China
JUNYUAN LIU, China University of Mining & Technology, Beijing, China
BO-WEN ZHANG*, University of Science and Technology Beijing, China

Despite recent advancements in semantic representation driven by pre-trained and large-scale language models, addressing long tail challenges in multi-label text classification remains a significant issue. Long tail challenges have persistently posed difficulties in accurately classifying less frequent labels. Current approaches often focus on improving text semantics while neglecting the crucial role of label relationships. This paper introduces LabelCoRank, a novel approach inspired by ranking principles. LabelCoRank leverages label co-occurrence relationships to refine initial label classifications through a dual-stage reranking process. The first stage uses initial classification results to form a preliminary ranking. In the second stage, a label co-occurrence matrix is utilized to rerank the preliminary results, enhancing the accuracy and relevance of the final classifications. By integrating the reranked label representations as additional text features, LabelCoRank effectively mitigates long tail issues in multi-label text classification. Experimental evaluations on popular datasets including MAG-CS, PubMed, and AAPD demonstrate the effectiveness and robustness of LabelCoRank. The implementation code is publicly available on https://github.com/821code/LabelCoRank.

JAIR Associate Editor: Kai-Wei Chang

JAIR Reference Format:
Yan Yan, Junyuan Liu, and Bo-Wen Zhang. 2025. LabelCoRank: Revolutionizing Long Tail Multi-Label Classification with Co-Occurrence Reranking. *Journal of Artificial Intelligence Research* 83, Article 14 (July 2025), 19 pages. DOI: 10.1613/jair.1.16968

1 Introduction

Multi-label text classification (MLTC) is a crucial task in natural language processing (NLP) [3]. Its goal is to assign appropriate labels to a given text. The exponential growth of textual information on the internet [7] has made MLTC increasingly important for extracting valuable insights from vast amounts of text. One of the most prominent examples highlighting the necessity of MLTC is the field of scientific research. The number of academic publications is growing exponentially each year, making it difficult for researchers, librarians, and digital platforms to efficiently organize and manage this vast body of knowledge. Efficient and cost-effective management of these papers is an urgent problem that needs to be addressed. In particular, the biomedical field faces significant challenges in literature indexing and retrieval. In the biomedical article retrieval field, MEDLINE, a major component of PubMed maintained by the National Library of Medicine (NLM), uses Medical Subject Headings (MeSH) for indexing. However, MeSH indexing is manually curated, making the process time-consuming and expensive. New articles often take 2 to 3 months to be indexed, with each article costing around $10 [23]. Essentially, this task can be categorized as multi-label text classification, where each article is assigned several labels. Additionally, MLTC tasks have wide-ranging applications in other fields such as recommendation systems [13, 27, 35, 36], sentiment analysis [9, 14], anomaly detection [2, 10], rumor detection [5], and question-answering tasks [24].

*Corresponding Author.

Authors' Contact Information: Yan Yan, ORCID: 0000-0002-0187-7010, yanyan@cumtb.edu.cn, China University of Mining & Technology, Beijing, Beijing, China; Junyuan Liu, ORCID: 0009-0002-3434-3004, liujunyuan.me@foxmail.com, China University of Mining & Technology, Beijing, Beijing, China; Bo-Wen Zhang, ORCID: 0000-0002-6384-2104, bowenzhang@ustb.edu.cn, University of Science and Technology Beijing, Beijing, China.

DOI: 10.1613/jair.1.16968

1.1 Current Situation and Challenges

However, MLTC faces significant challenges, particularly the long-tail problem. In many datasets, a small number of labels appear frequently (head labels), while the majority of labels are infrequent (tail labels). This long-tail distribution makes it difficult to train effective classifiers, as tail labels lack sufficient training samples. Accurately classifying these less frequent labels remains a significant issue.

Traditional machine learning methods like KNN [28], SVM [8], logistic regression [26], and decision trees [15] often use feature representations such as Bag-of-Words (BOW) and N-Grams. While these methods have been foundational, they struggle to capture contextual information effectively, leading to limitations in handling complex text data. These traditional methods typically rely on simplistic representations that do not account for the intricate relationships between words and their contexts, resulting in poor performance, especially for rare labels.

Recent advancements in deep learning have led to the development of models based on CNNs and RNNs [4, 18, 19], which have demonstrated superior feature extraction capabilities. However, these models face challenges in capturing the contextual relationships of words in long texts [1]. For instance, CNNs, while effective in capturing local patterns, struggle with long-range dependencies. RNNs, though better at handling sequential data, often suffer from issues like vanishing gradients, making it hard to learn long-term dependencies effectively. The introduction of attention mechanisms has alleviated some of these issues, with models combining attention mechanisms to improve semantic representations of text and classification performance [40, 43, 44]. Transformer-based models [30] and powerful pre-trained models such as BERT [6], RoBERTa [20], and others have further enhanced text feature extraction capabilities. These models use self-attention mechanisms to capture relationships between words in a sequence, enabling better handling of long-range dependencies and improving the overall semantic understanding of texts.

Despite these advancements, current models primarily focus on text features and often neglect the relationships among labels and between labels and text. In certain scenarios, relying solely on text features is insufficient for predicting the correct labels, particularly when the text features cannot adequately represent the semantic meaning of the labels. This issue is especially pronounced for tail labels, which lack sufficient training samples and thus are more challenging to associate with text features. For example, when classifying scientific papers based on their abstracts, the short text of the abstracts may not fully convey the semantics of all labels, or the text may be expressed in a more obscure and ambiguous manner. In such cases, it is necessary to utilize the relationships among labels to identify tail labels that are not directly evident from the text semantics. For instance, in medical document classification, the presence of one disease label might imply the likelihood of another co-occurring disease. However, many existing models fail to capture these relationships adequately. Some recent works have tried to incorporate label information into the classification process. For instance, some have used attention mechanisms to explore the relationship between label semantics and text [29, 37]. However, they have not considered the relationships among the labels. Other researchers have employed graph networks to model labels and explore the relationships among labels [17, 22, 25, 31, 45]. Yet, in these studies, the expression of the same label information is consistent across different samples, neglecting the fact that label information should exhibit varying importance for different texts.

1.2 Our Contributions

To address these issues, we propose LabelCoRank, a novel approach inspired by ranking principles. LabelCoRank leverages label co-occurrence relationships to refine initial label classifications through a dual-stage reranking process. The first stage uses initial classification results to form a preliminary ranking. In the second stage, a label co-occurrence matrix is utilized to rerank the preliminary results, enhancing the accuracy and relevance of the final classifications. By integrating the reranked label representations as additional text features, LabelCoRank effectively mitigates long-tail issues in multi-label text classification. The model also incorporates the frequency distribution of labels, allowing it to assign varying importance to labels based on their occurrence in the dataset. Through label embedding and attention mechanisms, LabelCoRank establishes semantic relationships between labels and text features. This dual-stage approach ensures that even infrequent labels, which are typically underrepresented, receive adequate attention during the classification process.

The contributions of this paper are:

- We propose LabelCoRank, leveraging pre-trained models and incorporating label co-occurrence and label-text semantic relationships to improve label prediction accuracy.

- We introduce the frequency distribution of labels into the prediction process, allowing varying importance of labels across different samples. The model employs three rounds of label enhancement, utilizing pre-trained models, label co-occurrence matrices, and attention mechanisms based on label position embeddings.
- Experimental results on MAG-CS, PubMed, and AAPD datasets demonstrate the model's effectiveness and robustness, validating its suitability for both large-scale and small-scale label datasets.

2 Related Work

2.1 Multi-label Text Classification

Deep learning has become the mainstream method for addressing the multi-label text classification problem. TextCNN [4] pioneered the use of convolutional networks to capture local features, integrating these features into text representations through global pooling operations. XML-CNN[18] designed a dynamic pooling module to capture richer information from different regions of a document. Compared to convolutional neural networks, recurrent neural networks and attention mechanisms have shown superior performance in capturing contextual associations within texts, making them particularly effective for multi-label text classification tasks. LSAN[37] utilizes label semantic information to construct label-specific document representations and employs a self-attention mechanism to measure the contribution of each word to each label. HAN[43] applies attention mechanisms at both the word and sentence levels, ultimately aggregating these into a text vector to obtain richer text features. AttentionXML[44] uses Bi-LSTM(Bidirectional Long Short-Term Memory) to capture long-distance dependencies between words and employs a multi-label attention mechanism to identify the most important parts of the text for each label. Transformer-based models have demonstrated even more powerful feature extraction capabilities, giving rise to a series of widely used pre-trained models such as BERT[6], RoBERTa[20], and XLNet[42].Star-Transformer[11] reduces the complexity compared to standard Transformer while retaining the ability to capture local components and long-range dependencies. BERTXML[39] introduces the attention mechanism to predict labels based on BERT, achieving good effect improvement. MATCH[47], built on the Transformer architecture, introduces metadata embeddings and uses linear layers to capture their relationship with the text.

However, these methods primarily focus on text feature representation, often neglecting label information. Recently, researchers have begun exploring how to better utilize label information. CorNet[39] adds additional linear layer combinations to create the CorNet module, which learns the correlations between labels and can be easily integrated into other models. GUDN[33] directly leverages pre-trained models to represent both label semantics and text semantics, using contrastive learning to explore the relationship between labels and text.On the other hand,graph-based methods have emerged as a significant approach to utilizing label information. MAGNET[25] uses label embeddings as node features and employs graph attention networks to learn dependencies between labels. LR-GCN[31] designs a method that utilizes external information from Wikipedia for label embedding, aiming to explore label dependencies and semantics.LDGN[22] combines the statistical label collaboration graph and the dynamic reconstruction graph to form a dual GCN, so as to learn the high-order relationship between labels. LiGCN[17] builds a graph for labels and text tokens, and strengthens the edge weight between specific tokens and the corresponding label through the graph convolutional network(GCN). S-GCN[45] combines text, words, and labels to construct a global graph, considering both semantic associations and global word relationships.By focusing on label information, these methods aim to enhance the multi-label text classification performance, addressing the limitations of traditional text-focused approaches.

2.2 Label Relationship

In multi-label text classification tasks, the labels of the data set will show different correlations. Different models have different methods of utilizing label relationships. Properly utilizing label relationships is an important research direction. Currently, there are three ways to construct label relationships.

The first direction is to establish label relationships based on the co-occurrence frequency matrix, which is the most intuitive, convenient and commonly used. LACO[46]. constructs two matrices: one is a pairwise label co-occurrence matrix, with 0 and 1 representing infrequent and frequent label co-occurrence respectively, and the other matrix is a conditional label co-occurrence matrix, which indicates whether other labels will co-occur when some labels appear. CL-MIL[16] will divide labels into three constraint relationships: direct correlation, indirect correlation, and irrelevant. It more fully reflects the relevance strength of the label.

The second direction is to calculate the cosine similarity of label embeddings. When selecting similar labels of a certain label in TailMix[12], the cosine similarity between all one-hop neighbor nodes in the label graph and this node is calculated, and the most similar node is selected.

The third direction is to extract the parent-child relationship between labels. Sometimes,the co-occurrence frequency between labels cannot directly reflect the deep relationship between labels. KenMesh[34] uses a two-layer GCN network to create a graph for the parent label and the child label respectively to reflect the parent-child relationship. However, only considering the parent-child vertical relationship between labels will ignore the horizontal relationship between labels. LGCCN[38] can capture the vertical dependencies between label levels, model the horizontal correlations, and construct a tree-like label graph without the need for the data set to provide label relationships in advance.

3 Methodology

In this section, we will first briefly introduce the overall algorithm flow of LabelCoRank (Section 3.1), followed by a detailed explanation of the implementation and motivation of the key component in the model, the Label Reranking with Co-occurrence Relationship method (Section 3.2).

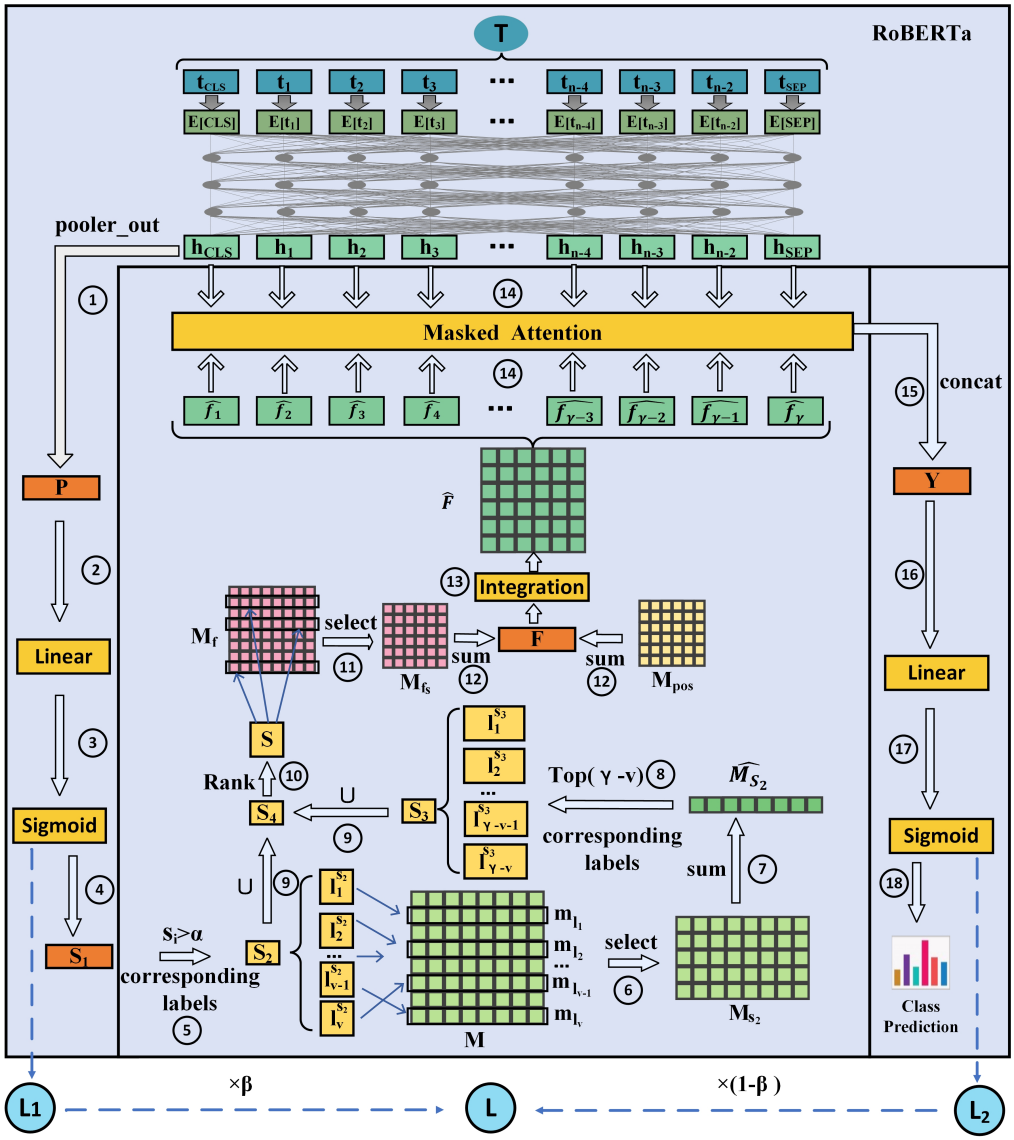

Fig. 1. The architecture of LabelCoRank

3.1 The Whole LabelCoRank Pipeline

The LabelCoRank model consists of three main modules. The first module uses text features captured by a pre-trained model to obtain an initial classification ranking. In the second module, by leveraging the head labels from the initial classification and combining them with the label frequency co-occurrence matrix and label frequency distribution information from the dataset, Label Reranking is performed. This results in a label feature sequence that incorporates various additional information, which is then used to establish semantic relationships with the text through an attention mechanism. The third module uses these features for the final classification. Fig. 1 illustrates the overall architecture of the LabelCoRank model. The following three paragraphs will provide a detailed introduction to each of these modules. The numbers from 1 to 18 in the Fig. 1 indicate the order of the method to help understand the flow of the data.

Initially, the RoBERTa model is employed for feature extraction and initial label prediction based on its text features. The text T is composed of words t, the initial text undergoes tokenization and other preprocessing steps, augmented with special tokens [CLS] and [SEP], resulting in a sequence of n words, $T = \{t_{cls}, t_1, t_2, t_3, \ldots, t_{n-2}, t_{sep}\}$. RoBERTa encodes this text, with the output of its last hidden layer denoted as $H = \{h_{cls}, h_1, h_2, h_3, \ldots, h_{n-2}, h_{sep}\}$. h_{cls} represents the output of the hidden layer after encoding the [CLS] token and can also be viewed as the representation of the text features after encoding with RoBERTa. The pooled representation of the text is denoted as P, formulated as follows:

$$H = RoBERTa(W_{RoBERTa}, T) \tag{1}$$

$$P = Tanh(h_{cls}W_1 + b_1) \tag{2}$$

The full-text features P encoded by RoBERTa are used for classification, and the sequence of label probabilities is denoted as $S_1 = \{s_1, s_2, \ldots s_K\}$, with K is the number of all labels.

$$S_1 = sigmoid(PW_2 + b_2) \tag{3}$$

For loss computation, the Binary Cross-Entropy Loss function is used, formulated as follows:

$$L_1 = -\frac{1}{K}\sum_{i=1}^{K}(\hat{s}_i \log(s_i) + (1 - \hat{s}_i)\log(1 - s_i)) \tag{4}$$

where K is the number of all labels, $\hat{s}_i$ is the true value of the i-th label, s_i is the predicted probability value of the i-th label. The prediction loss for this instance is denoted as L_1, which will be combined with the final classification loss L_2 as the ultimate loss L. The motivation for using RoBERTa, a widely-used feature extraction module in NLP that has demonstrated excellent performance, extends to its high accuracy in predicting head labels, although it performs less effectively on tail labels. This discrepancy has inspired us to improve its capability in predicting tail labels while leveraging its strong performance on head labels to use predicted head label information effectively.

Subsequently, the prediction results are combined with the label frequency correlation matrix M to obtain an extended label sequence, incorporating more relevant labels. This is regarded as the first Label Reranking. The sequence is sorted by frequency to obtain an ordered label sequence S that includes frequency information. This label sequence is mapped to label features, integrating positional information, and further fitting it through a linear layer. This results in the fused label features $\widehat{F} = \{\widehat{f_1}, \widehat{f_2}, \widehat{f_3}, \ldots, \widehat{f_Y}\}$, allowing the same label to have different feature representations at different positions in the sequence. This is considered the second Label Reranking. The label feature sequence is then used to perform masked attention learning on the character-level features extracted by RoBERTa, ultimately obtaining features that incorporate expanded label information. The next section will explain how the final label feature sequence containing more relevant information is achieved through Label Reranking, why this process is referred to as Label Reranking, and the motivations behind this approach.

Finally, these features are used for classification. The features that incorporate expanded label information extracted through the multi-head masked attention mechanism, MultiHead$(\widehat{f}, H)$, where $\widehat{f} \in \widehat{F}$, are concatenated to obtain the final feature f_{cat}, as follows:

$$f_{cat} = Concat(MultiHead(\widehat{f}, H)) \tag{5}$$

The features are mapped to the corresponding probabilities of K labels through a linear transformation and a sigmoid function.

$$Y = sigmoid(f_{cat}W_5 + b_5) \tag{6}$$

The binary cross-entropy loss function for $Y = \{y_1, y_2, \ldots y_K\}$ is used as the loss function for the second prediction, resulting in the loss L_2. The final loss is defined as:

$$L_2 = -\frac{1}{K}\sum_{i=1}^{K}(\hat{y}_i \log(y_i) + (1-\hat{y}_i)\log(1-y_i)) \tag{7}$$

$$L = \beta \times L_1 + (1-\beta) \times L_2 \tag{8}$$

where K is the number of all labels, $\hat{y}_i$ is the true value of the i-th label, y_i is the predicted probability value of the i-th label and β is a hyperparameter used to balance the influence of the two losses on the final result.

3.2 Label Reranking with Co-occurance Relationship

In the first stage, an initial ranking is formed using the preliminary classification results. In the second stage, we re-rank these results to integrate various additional information beyond text features, helping to address the long-tail problem in classification. This section will specifically introduce the method of Label Reranking with Co-occurrence Relationship.

After the initial prediction, the first predicted label probability sequence S_1 is obtained. A threshold hyperparameter α is used to select the parts of S_1 where the label prediction probabilities are greater than α, and the corresponding labels are taken as the label sequence S_2. This aims to obtain labels more relevant to the text as predicted by RoBERTa, thereby reducing the introduction of irrelevant noise.

For the co-occurrence frequency matrix M, its contents are defined as follows: for a dataset with K labels, $m_{i,j}$ represents the frequency with which the j label appears when the i label is present. m_i is the co-occurrence frequency sequence of all labels corresponding to the i label, that is, $m_i = \{m_{i,1}, m_{i,2}, ..., m_{i,k}\}$. Therefore, for the label sequence S_2 with V labels, $S_2 = \{l_1, l_2, \ldots, l_v\}$, the label frequency sequence set is $M_{S_2} = \{m_{l_1}, m_{l_2}, \ldots, m_{l_v}\}$. Summing the label frequency sequences yields $\widehat{M_{s_2}}$, defined as follows:

$$\widehat{M_{s_2}} = \sum_{v=1}^{V} m_{l_v} = \left\{\sum_{v=1}^{V} m_{l_v,1}, \sum_{v=1}^{V} m_{l_v,2}, \ldots, \sum_{v=1}^{V} m_{l_v,k}\right\} \tag{9}$$

A hyperparameter γ is used to control the desired length of the label sequence. The corresponding labels from $\widehat{M_{s_2}}$ are obtained, and the top $\gamma - V$ labels that are not part of S_2 are selected based on frequency in descending order to form the label sequence S_3. $S_2 \cup S_3$ is used as the label sequence S_4, which is more relevant to each sample:

$$S_4 = S_2 \cup S_3 \tag{10}$$

This is regarded as the first Label Reranking. In terms of effectiveness, using the co-occurrence frequency matrix allows samples to expand to include more relevant labels, thereby obtaining more related information. However, from another perspective, through the label co-occurrence matrix, important information in the initial label probability sequence that might have been overlooked is reordered and reprioritized based on co-occurrence information.

S_4 is then sorted based on the label frequency distribution across the entire dataset, placing high-frequency labels at the beginning of the sequence and low-frequency labels at the end. This results in a frequency-integrated label sequence S:

$$S = Rank(S_4) \tag{11}$$

A trainable label position feature matrix $M_{\text{pos}} \in \mathbb{R}^{\gamma \times \delta}$ and a label feature matrix $M_{\text{f}} \in \mathbb{R}^{K \times \delta}$ are designed, where δ is the size of the RoBERTa hidden layer. Using S, the corresponding label features from M_{f} are selected and added to M_{pos} to obtain the feature representation F with integrated label position information. A feedforward neural network module is then used to further integrate the position information and label information in F, represented as follows:

$$\hat{F} = ReLU(Drop(FW_3 + b_3))W_4 + b_4 \tag{12}$$

This is considered the second Label Reranking. By sorting labels according to their distribution in the dataset, label frequency distribution information is incorporated, and all relevant labels are re-ranked. This effectively organizes the highly semantically relevant head labels initially predicted by RoBERTa and the highly correlated labels obtained through co-occurrence relationships.

The design has two main motivations. First, by reranking the labels based on the frequency distribution in the dataset, high-frequency labels are placed at the beginning of the sequence. This way, the selected label

feature information sequence also contains the importance information of frequency. During subsequent network training using label feature information, the neural network gives different levels of attention to the information at the beginning and end of the sequence, thus integrating this information into the model. Second, the position feature matrix is designed because different samples have different label sequences S, and the order of S contains label frequency information. For the same label, its meaning should be different depending on its position in different label sequences. After sorting and filtering, the labels that appear earlier in the sequence are more important, whereas the labels that appear later are less important. Therefore, the same label in different positions within the sequence supplements the corresponding label with feature information specific to its position.

In the end, the label feature sequence $\widehat{F} = \{\widehat{f_1}, \widehat{f_2}, \ldots, \widehat{f_\gamma}\}$, combined with a multi-head masked attention mechanism, is used to extract information from the RoBERTa-encoded text that is more relevant to the label information.

$$MultiHead(\widehat{f}, H) = Concat(head_1, head_2, head_3 ... head_\eta) \tag{13}$$

$$head_i = Attention(Q_i, K_i, V_i) = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}} + M)V_i \tag{14}$$

where $Q_i = \widehat{f} W_i^Q, K_i = HW_i^K, V_i = HW_i^V \quad for\ i \in [1, \eta], \widehat{f} \in \widehat{F}$. η is the number of heads . W_i^Q, W_i^K, and W_i^V are the matrices parameters to be learned for the i-th head. d_k is the dimension of the keys, and M is the attention mask that is added to the scores.

The above outlines the entire process of Label Reranking with Co-occurrence Relationship. Its purpose is to integrate various additional label information through Label Reranking to address the long-tail problem in multi-label text classification. In the following comparative and ablation experiments, we will explore the effectiveness of this method and verify the design of LabelCoRank.

4 Experiments

4.1 Datasets

The proposed method is evaluated on three well-known public datasets: MAG-CS, PubMed, and AAPD. The reasons for selecting these datasets are as follows.

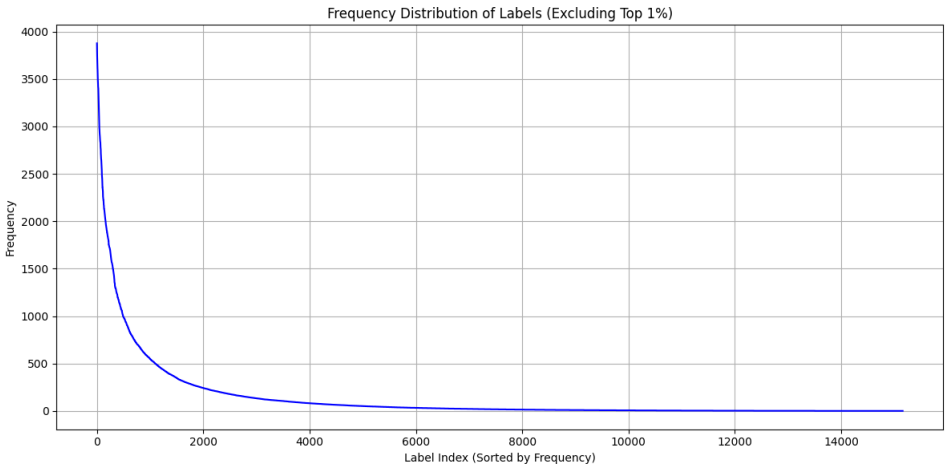

Fig. 2. Frequency Distribution of Labels on MAG-CS

First, both PubMed and MAG-CS exhibit clear long-tail label distributions. As shown in Fig. 2 and Fig. 3, the label frequency distributions of PubMed and MAG-CS are highly imbalanced, characterized by a small number of frequent labels and a large number of rare labels (the top 1% of labels are not displayed in the figures to avoid making the graphs overly "L"-shaped). Similarly, the label frequency distribution of AAPD, shown in Fig. 4, also presents a long-tail pattern. The presence of a pronounced long-tail distribution is crucial for evaluating methods aimed at improving tail label prediction.Second, PubMed, MAG-CS, and AAPD are all datasets originating from the scientific and academic domains, where label systems are more standardized and annotation noise is relatively

low. This provides a stable and reliable environment for accurately evaluating the model's capability to enhance tail label predictions.

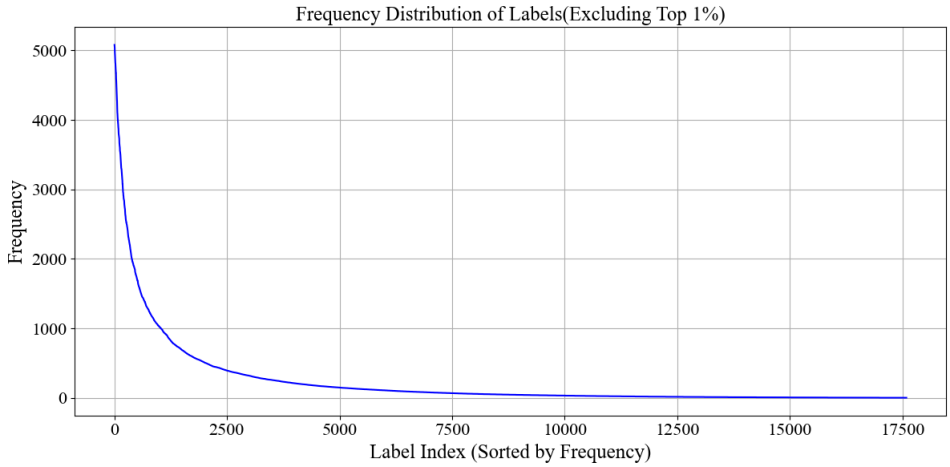

Fig. 3. Frequency Distribution of Labels on PubMed

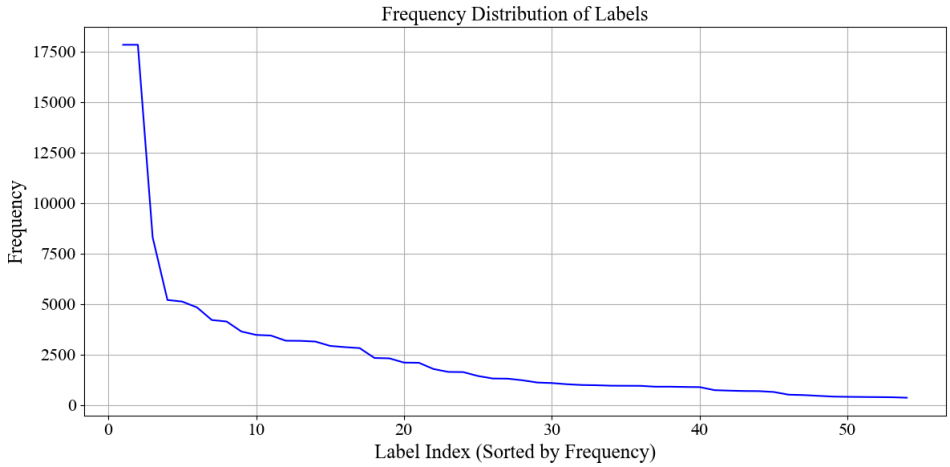

Fig. 4. Frequency Distribution of Labels on AAPD

Table 1. Dataset statistics.N_{trn} and N_{tst} represent the number of documents in the training set and test set, respectively. D represents the vocabulary size of all documents. L represents the number of labels. L_{avg} represents the average number of labels per document, and W_{avg} represents the average number of words per document.

Datasets	N_{trn}	N_{tst}	D	L	L_{avg}	W_{avg}
MAG-CS	564340	70534	425345	15809	5.60	126.33
PubMed	718837	89855	776975	17963	7.78	198.97
AAPD	54840	1000	69399	54	2.41	163.43

The datasets are introduced as follows, and Table 1 presents the statistics of these three datasets.

MAG-CS[32]: The dataset consists of 705,407 papers from the Microsoft Academic Graph (MAG), selected from 105 prominent CS conferences held between 1990 and 2020. It comprises 15,808 unique labels, providing a comprehensive collection of scientific literature in the field.

PubMed[21]: The dataset comprises 898,546 papers sourced from PubMed, representing 150 leading medical journals from 2010 to 2020. It includes 17,963 labels corresponding to MeSH terms, offering valuable insights into biomedical research.

AAPD[41]: The dataset comprises English abstracts of computer science papers sourced from arxiv.org, each paired with relevant topics. In total, it includes 55,840 abstracts spanning various related disciplines.

4.2 Baselines and Evaluation Metrics

4.2.1 Baselines.

The following comparison methods were selected. They are categorized into three types: methods based on traditional neural networks, methods based on Transformers, and methods similar to ours that also utilize pre-trained models and incorporate label information.

- **XML-CNN**[18] utilizes a dynamic max pooling scheme to capture richer information from different regions of the document, adopts a binary cross-entropy loss function to handle multi-label problems, and introduces hidden bottleneck layers to obtain better document representation and reduce model size.
- **MeSHProbeNet**[40] is an end-to-end deep learning model designed for MeSH indexing, which assigns MeSH terms to MEDLINE citations. It won the first place in the latest batch of Task A in the 2018 BioASQ challenge.
- **AttentionXML**[44] is a label tree-based deep learning model designed for Extreme Multi-label Text Classification (XMTC). It introduces two key features: a multi-label attention mechanism that captures the relevant text parts of each label, and a shallow and wide probabilistic label tree that can efficiently handle millions of labels.
- **Transformer**[30] is a network architecture, the first sequence transformation model based entirely on the attention mechanism. It replaces the commonly used cyclic layer with a multi-head self-attention mechanism and abandons the common cyclic and CNN structures.
- **Star-Transformer**[11] is a lightweight alternative to Transformer for NLP tasks. It uses a star topology to reduce the complexity from quadratic to linear and solve the problem of heavy computing requirements.
- **BertXML**[39] is a customized version of BERT designed specifically for XMTC. It overcomes the limitation of BERT's single [CLS] token by incorporating multiple [CLS] tokens at the start of each input sequence.
- **MATCH**[47] learns improved text and metadata representations by jointly embedding them in the same space, enabling higher-order interactions between words and metadata.
- **LiGCN**[17] introduced an interpretable graph convolutional network model that models tokens and labels as nodes in a heterogeneous graph. It calculates the cosine similarity between label embeddings to capture the relationships among labels.
- **GUDN**[33] leverages label semantics and deep pre-trained models, incorporating a label reinforcement strategy for fine-tuning to enhance classification performance. The model exhibits sensitivity to label semantics and shows remarkable efficacy on datasets with semantically rich labels.

4.2.2 Evaluation Metrics.

In multi-label text classification tasks, commonly used evaluation metrics include Precision, mAP, Recall, Accuracy, F1, P@K, and NDCG@K. In this study, P@K and NDCG@K are selected because they are better suited for datasets with large label spaces and severe sparsity, such as PubMed and MAG-CS(see Table 1). These metrics avoid the bias caused by the overwhelming number of negative labels in traditional metrics and focus on the quality of Top-K predictions. Additionally, NDCG@K considers the ranking of predicted labels. Using these metrics also ensures comparability with previous work and maintains consistency in experimental evaluation. We follow the research by Zhang et al.[47], adhering to two rank-based evaluation metrics: precision at top k (P@k) and normalized Discounted Cumulative Gain at top k (NDCG@k), where $k = 1, 3, 5$. For a document d, let $\boldsymbol{y}_d \in \{0,1\}^{|\mathcal{L}|}$ be its ground truth label vector and rank(i) be the index of the i-th highest predicted label according to the re-ranker.

$$P@k = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{y}_{d,rank(i)} \tag{15}$$

$$DCG@k = \sum_{i=1}^{k} \frac{\boldsymbol{y}_{d,rank(i)}}{\log_2(i+1)} \tag{16}$$

$$NDCG@k = \frac{DCG@k}{\sum_{i=1}^{\min(k,||\boldsymbol{y}_d||_0)} \frac{1}{\log_2(i+1)}} \tag{17}$$

4.3 Experimental Settings

All experimental environments were consistent. Experiments were conducted on a computer equipped with an Nvidia RTX 4090 GPU and 128 GB of RAM. All comparative methods using pre-trained models, as well as our approach, utilized the RoBERTa pre-trained model as a feature extractor. The hyperparameters for all comparative methods were set according to their original papers or code. The AdamW optimizer was employed with a learning rate of 1e-5, sentence truncation set to 512, and batch size of 16. The threshold hyperparameter, α, was set to 0.3, and the hyperparameters for the loss function weight, β, was set to 0.3, 0.3, and 0.25 for the MAG-CS, PubMed, and AAPD datasets, respectively. The number of selected labels for these datasets γ was 30, 35, and 20, respectively.

4.4 Experimental Results

4.4.1 Performance on The Complete Datasets.

Tables 2, 3, and 4 summarize the results of different models on the MAG-CS, PubMed, and AAPD datasets, respectively.

In the MAG-CS dataset (Table 2), the model outperforms all baselines on almost all metrics, except for P@1, where it lags behind MATCH by 0.0072. However, for P@3, P@5, NDCG@3 and NDCG@5, the model improves over MATCH by 0.0047, 0.0131, 0.0007 and 0.0079, respectively. This indicates an enhanced prediction performance for tail labels. This improvement is attributed to the introduction of a significant amount of relevant label information, which boosts the prediction of tail labels but slightly dilutes the focus on head labels. Additionally, MATCH utilizes extra meta-information that LabelCoRank does not, which could explain the differences for P@1.

Table 2. Results on MAG. Following the experiments from the MATCH[47], the average of three trials was taken for comparison. The best experimental result on the MAG dataset is indicated in bold.

Model	P@1	P@3	P@5	NDCG@3	NDCG@5
XML-CNN[47]	0.8656	0.7028	0.5756	0.7842	0.7407
MeSHProbeNet[47]	0.8738	0.7219	0.5927	0.8020	0.7588
AttentionXML[47]	0.9035	0.7682	0.6441	0.8489	0.8145
Transformer[47]	0.8805	0.7327	0.6024	0.8129	0.7703
Star-Transformer[47]	0.8569	0.7089	0.5853	0.7876	0.7486
BERTXML[47]	0.9011	0.7532	0.6238	0.8355	0.7954
MATCH[47]	**0.9190**	0.7763	0.6457	0.8610	0.8223
RoBERTa-LiGCN	-	-	-	-	-
GUDN	0.8757	0.7354	0.6173	0.8137	0.7805
LabelCoRank	0.9118	**0.7810**	**0.6588**	**0.8617**	**0.8302**

In the PubMed dataset (Table 3), LabelCoRank achieves the best performance across all metrics. It is worth noting that LabelCoRank demonstrates a significantly better performance than MATCH on the PubMed dataset compared to the MAG dataset. The improvements in P@3, P@5, NDCG@3, and NDCG@5 all exceed 2.8 percentage points, suggesting a substantial improvement in predicting difficult-to-predict tail labels. The PubMed dataset has the highest average number of labels per instance, which likely enhances the relevance of supplementary label information, leading to superior performance.

In the AAPD dataset (Table 4), LabelCoRank also achieves the best performance across all metrics. Despite the average number of labels per document being only 2.41, the total number of labels is 54, allowing most label features to participate in the model's computations. LabelCoRank effectively learns and distinguishes relevant from irrelevant label information without introducing excessive noise from additional labels. This demonstrates the model's robustness even in tasks with fewer labels.

From these experiments, it is evident that models based on the Transformer architecture outperform traditional CNN and RNN-based models. When comparing models that also use pre-trained models and incorporate label information, namely RoBERTa-LiGCN and GUDN, LabelCoRank consistently outperforms them. GUDN's

Table 3. Results on PubMed. Following the experiments from the MATCH[47], the average of three trials was taken for comparison. The best experimental result on the PubMed dataset is indicated in bold.

Model	P@1	P@3	P@5	NDCG@3	NDCG@5
XML-CNN[47]	0.9084	0.7182	0.5857	0.7790	0.7075
MeSHProbeNet[47]	0.9135	0.7224	0.5878	0.7836	0.7109
AttentionXML[47]	0.9125	0.7414	0.6169	0.7979	0.7341
Transformer[47]	0.8971	0.7299	0.6003	0.7867	0.7178
Star-Transformer[47]	0.8962	0.6990	0.5641	0.7612	0.6869
BERTXML[47]	0.9144	0.7362	0.6032	0.7949	0.7247
MATCH[47]	0.9168	0.7511	0.6199	0.8072	0.7395
RoBERTa-LiGCN	-	-	-	-	-
GUDN	0.9241	0.7760	0.6524	0.8285	0.7683
LabelCoRank	**0.9243**	**0.7794**	**0.6571**	**0.8315**	**0.7729**

Table 4. Results on AAPD. Following the experiments from the LiGCN[17], the best experimental results were taken for comparison. The best experimental result on the AAPD dataset is indicated in bold.

Model	P@1	P@3	P@5	NDCG@3	NDCG@5
XML-CNN	0.8140	0.5903	0.4064	0.7771	0.8187
MeSHProbeNet	0.8460	0.6020	0.4130	0.7969	0.8384
AttentionXML	0.8500	0.6110	0.4180	0.8081	0.8483
Transformer	0.7820	0.5687	0.3930	0.7456	0.7853
Star-Transformer	0.7790	0.5697	0.3976	0.7471	0.7917
BERTXML	0.8140	0.5847	0.3972	0.7737	0.8093
MATCH	-	-	-	-	-
RoBERTa-LiGCN[17]	0.8250	0.6126	0.4138	0.8039	0.8383
GUDN	0.8260	0.6023	0.4136	0.7911	0.8335
LabelCoRank	**0.8540**	**0.6227**	**0.4188**	**0.8188**	**0.8526**

performance on the MAG dataset is weaker than on PubMed, likely because it relies directly on pre-trained models to extract label features, which are presented as IDs in MAG, making it difficult to capture label text features effectively. However, LabelCoRank allows the model to learn the feature representation of each label independently, without relying on label text features.

4.4.2 Performance Comparison of Head and Tail Labels.
To comprehensively verify the practical effectiveness of LabelCoRank in improving tail label prediction, we designed a focused experiment using two specifically constructed subsets from the PubMed dataset. The first subset is the head label dataset, which consists of samples where the real labels are entirely formed by the top 10% most frequent head labels, totaling 226,629 samples. The second subset is the tail label dataset, which consists of samples where the top labels appear no more than twice in the real labels, totaling 93,364 samples. Fig. 3 shows the label distribution of the PubMed dataset after excluding the top 1% frequent labels (the distribution of the top 1% labels is not displayed in the figure to prevent the graph from becoming too "L"-shaped). The top 10% most frequent labels are considered head labels, while the remaining 90% are considered tail labels. Since the dataset consisting entirely of tail label samples only contains 3,803 samples, we selected a subset consisting of samples where the top labels appear no more than twice in the real labels.

On the head label dataset (Table 5), LabelCoRank outperforms the current state-of-the-art models across all evaluation metrics.LabelCoRank shows an improvement of nearly 0.6% and 0.9% over the second-best method in P@3 and P@5, respectively, and also achieves gains in the NDCG metrics. On the tail label dataset (Table 6), the advantage of LabelCoRank is more pronounced. Compared to other models, LabelCoRank achieves the best results in P@1 (0.8382), P@3 (0.5593), P@5 (0.3940), NDCG@3 (0.6584), and NDCG@5 (0.6431). For instance, P@3 and P@5 are improved by approximately 2.9% and 2.0% compared to GUDN, while NDCG@3 and NDCG@5

Table 5. Results on PubMed head label dataset

Model	P@1	P@3	P@5	NDCG@3	NDCG@5
XML-CNN	0.8692	0.6318	0.4914	0.7409	0.6969
MeSHProbeNet	0.9168	0.7183	0.5665	0.8233	0.7806
AttentionXML	0.9122	0.7111	0.5680	0.8155	0.7784
Transformer	0.8686	0.6371	0.4914	0.7426	0.6872
Star-Transformer	0.8709	0.6341	0.4915	0.7404	0.6871
RoBERTa	0.8960	0.6633	0.5162	0.7738	0.7284
BERTXML	0.8854	0.6606	0.5116	0.7675	0.7199
MATCH	-	-	-	-	-
RoBERTa-LiGCN	-	-	-	-	-
GUDN	0.9072	0.7040	0.5558	0.8089	0.7671
LabelCoRank	**0.9178**	**0.7240**	**0.5752**	**0.8284**	**0.7887**

Table 6. Results on PubMed tail label dataset

Model	P@1	P@3	P@5	NDCG@3	NDCG@5
XML-CNN	0.7509	0.4319	0.3017	0.5287	0.5142
MeSHProbeNet	0.7925	0.4927	0.3450	0.5910	0.5753
AttentionXML	0.7997	0.4878	0.3458	0.5856	0.5727
Transformer	0.7784	0.4640	0.3188	0.5638	0.5347
Star-Transformer	0.7625	0.4511	0.3122	0.5489	0.5227
RoBERTa	0.7739	0.4586	0.3141	0.5516	0.5278
BERTXML	0.8014	0.5213	0.3629	0.6170	0.5974
MATCH	-	-	-	-	-
RoBERTa-LiGCN	-	-	-	-	-
GUDN	0.8192	0.5306	0.3744	0.6279	0.6136
LabelCoRank	**0.8382**	**0.5593**	**0.3940**	**0.6584**	**0.6431**

are improved by around 3.0%. These results clearly demonstrate that LabelCoRank significantly improves the prediction accuracy for tail labels.

4.4.3 Comparison with GPT-4o.

In this section, we compare LabelCoRank with GPT-4o, a renowned advanced large language model, on the PubMed dataset. Due to the prompt length limitation, we are unable to provide the GPT-4o with all the labels (over 10,000 labels), thus preventing us from fully leveraging the capabilities of the GPT-4o. Instead, we use the GPT-4o to rerank the top ten predictions from RoBERTa and select the five most relevant labels. Viewed from another angle, this approach is a secondary enhancement of RoBERTa's predictions by GPT-4o, which is conceptually similar to what LabelCoRank does.

Table 7. Comparison with GPT-4o on the PubMed dataset

Model	P@1	P@3	P@5	NDCG@3	NDCG@5
RoBERTa*	0.9935	0.9046	0.7484	0.9474	0.8719
RoBERTa	0.9178	0.7343	0.6037	0.7941	0.7254
RoBERTa+GPT-4o	0.7990	0.6343	0.5304	0.6828	0.6271
LabelCoRank	**0.9243**	**0.7794**	**0.6571**	**0.8315**	**0.7729**

A template of the prompt used in the experiments is shown in Fig. 5. Additionally, we calculated the metric that indicates while all the correct labels be selected from RoBERTa's top ten predictions, which helps establish an upper bound to avoid the result being impacted by the lack of correct labels within RoBERTa's Top 10 predictions.

Prompt: Select the top five relevant labels from the given labels and return them as a Python list

Given the text '{data[count]}', and the following labels: {tags}, with corresponding indices {preList}, please select the five most relevant labels for the text and return the corresponding labels and indices in Python list format. For example: ['A', 'B', 'C', 'D', 'E'] and [1, 6, 3, 8, 4]. If there are fewer than 5 results, add the ones you find most relevant to make up the difference. No explanation is needed, just the relevant label list and its index list.\n

Fig. 5. Prompt Template.The content inside the {} brackets represents fillable fields, where {data[count]} refers to the text, {tags} refers to the labels, and {preList} refers to the indices corresponding to the labels.

We refer to it as RoBERTa*. The results after GPT-4o reranking were parsed and calculated, and the findings are in Table 7. The results show that our method outperforms GPT's selection based on RoBERTa's probability scores. Moreover, we found that GPT's performance actually deteriorated after reranking, which may suggest that the LLM struggles to adapt well to this task.

4.5 Ablation Study

From the analysis of the above experiments, it is evident that LabelCoRank effectively improves label prediction performance by incorporating label information, particularly enhancing the accuracy of predicting tail labels in datasets with a large number of labels. We conducted ablation experiments to verify the effectiveness of the modules used in LabelCoRank for integrating label information. Four design elements in LabelCoRank need to be validated for their effectiveness: label selection, label sequence ranking, incorporation of positional information, and selection of the number of labels.

4.5.1 Effects of Label Selection, Label Sequence Ranking and Position Information.

Label selection directly affects which information from the text will be extracted under the attention mechanism, thus impacting the final label prediction. To validate the effectiveness of using the frequency correlation matrix for label selection, a comparison was made against selecting the same number of labels directly from RoBERTa's predictions without the frequency correlation matrix. Experimental results demonstrate that label selection using the frequency correlation matrix is more effective than directly using RoBERTa's predicted labels. This is understandable, as the relevance of the predicted labels diminishes after a certain quantity. Additionally, RoBERTa's predicted labels do not account for co-occurrence relationships beyond textual semantics. For some labels, the connections are more latent and not directly reflected in the text's semantic content.

The label sequence is sorted based on the frequency of label occurrence across the entire dataset, aiming to introduce the frequency distribution information of labels over the dataset. When expanding the features obtained through the multi-label attention mechanism, this naturally positions high-frequency label features at the forefront, allowing the classifier to inherently pay different levels of attention to labels at different positions. This is because, for the classifier, each feature used for classification, from the head to the tail, consists of features extracted from high-frequency labels to those extracted from low-frequency labels. In comparison to an unordered feature sequence, where the contributions to classification are random, an ordered feature sequence allows for the anticipation that the features towards the front will contribute more. Therefore, as the classifier continuously assesses that the contribution from head labels is greater, the weights assigned to features from head labels will naturally be larger.

This label sequence, enriched with frequency distribution information, is considered a prioritized sequence. For the same label, its meaning varies depending on its position; the further forward it is, the more attention it should receive, and conversely, the further back it is, the less attention it should receive. Therefore, positional information was integrated into the label sequence, enabling different expressions at different positions to achieve dynamic label feature representation.

Experiments across three datasets validated the effectiveness of these approaches. Additionally, an ablation experiment was conducted by removing all the aforementioned designs to verify their overall effect as a reranking

Table 8. Results of ablation experiments

Dataset	Model	P@1	P@3	P@5	NDCG@3	NDCG@5
MAG-CS	w/o correlation matrix	0.9031	0.7680	0.6442	0.8483	0.8139
	w/o label ranking	0.9056	0.7725	0.6497	0.8527	0.8197
	w/o position information	0.9039	0.7692	0.6452	0.8498	0.8156
	w/o all reranking parts	0.8997	0.7578	0.6342	0.8402	0.8062
	LabelCoRank	**0.9118**	**0.7811**	**0.6591**	**0.8620**	**0.8305**
PubMed	w/o correlation matrix	0.9251	0.7700	0.6442	0.8243	0.7623
	w/o label ranking	**0.9287**	0.7761	0.6519	0.8298	0.7694
	w/o position information	0.9259	0.7699	0.6441	0.8242	0.7621
	w/o all reranking parts	0.9236	0.7608	0.6337	0.8165	0.7527
	LabelCoRank	0.9241	**0.7813**	**0.6593**	**0.8330**	**0.7749**
AAPD	w/o correlation matrix	0.8470	0.6193	**0.4204**	0.8129	0.8509
	w/o label ranking	0.8490	0.6193	0.4192	0.8145	0.8515
	w/o position information	0.8480	0.6160	0.4170	0.8138	0.8504
	w/o all reranking parts	0.8340	0.6132	0.4156	0.8077	0.8423
	LabelCoRank	**0.8540**	**0.6227**	0.4188	**0.8188**	**0.8526**

module. According to results presented in Table 8, both label sequence ordering and the integration of positional information significantly improved classification outcomes.

4.5.2 Effects of The Number of Labels.

Different choices in the number of labels introduce varying degrees of relevant label information and noise, impacting different datasets and samples differently. The effects of label number selection were investigated through experiments and analysis. Results(Table 9) indicate that in the MAG dataset, 35 labels were optimal; in the Mesh dataset, 30 labels were suitable ; and in the AAPD dataset, 20 labels were appropriate.

Table 9. Results with different number of labels. Number of labels refers to the size of the sequence of labels considered relevant in reranking.

Dataset	Number of labels	P@1	P@3	P@5	NDCG@3	NDCG@5
MAG-CS	40	0.9085	0.7767	0.6550	0.8572	0.8255
	35	0.9118	**0.7811**	**0.6589**	**0.8620**	**0.8305**
	30	**0.9125**	0.7797	0.6569	0.8606	0.8283
	25	0.9102	0.7802	0.6586	0.8606	0.8295
	20	0.9055	0.7722	0.6053	0.8527	0.8207
PubMed	35	0.9275	0.7805	0.6567	**0.8332**	0.7735
	30	0.9241	**0.7813**	**0.6593**	0.8330	**0.7749**
	25	0.9282	0.7772	0.6539	0.8306	0.7709
	20	0.9275	0.7765	0.6511	0.8300	0.7688
	15	**0.9293**	0.7759	0.6518	0.8299	0.7695
AAPD	30	0.8480	0.6130	**0.4212**	0.8102	**0.8531**
	25	0.8480	0.6183	0.4210	0.8131	0.8522
	20	**0.8540**	**0.6227**	0.4189	**0.8188**	0.8526
	15	0.8480	0.6177	0.4180	0.8149	0.8507
	10	0.8350	0.6117	0.4172	0.8041	0.8434

4.6 Case Study

From an information retrieval perspective, Our model can be viewed as a dual-prediction system from an information retrieval perspective. Initially, text features extracted by RoBERTa are used for preliminary label retrieval within the label space. This initial retrieval expands the relevant label sequence, generating a label information sequence related to the text. Subsequently, an attention mechanism extracts features from the text's word features that are more relevant to these label features for a secondary prediction. We demonstrate the effectiveness of our method by comparing it to the AttentionXML and BERTXML models on the PubMed dataset, illustrating the improvements introduced in the secondary retrieval phase.

Table 10. Case 1.Extended labels represent additional label information obtained through the correlation matrix. This can be used to analyze why the introduction of additional label information is effective.The prediction results are the top-5 labels predicted by each method.Green: Correct predictions. Red: Incorrect predictions. Orange:Semantics in the text that are directly related to labels.

Text	gb1 is not a two state folder identification and characterization of an on pathway intermediate the folding pathway of the small α β protein gb1 has been extensively studied during the past two decades using both theoretical and experimental approaches these studies provided a consensus view that the protein fold in a two state manner here we reassessed the folding of gb1 both by experiments and simulations and detected the presence of an on pathway intermediate this intermediate has eluded earlier experimental characterization and is distinct from the collapsed state previously identified using ultrarapid mixing failure to identify the presence of an intermediate affects some of the conclusions that have been drawn for gb1 a popular model for protein folding studies
Extend labels	'Humans', 'Animals', 'Protein Conformation', 'Protein Structure, Tertiary', 'Amino Acid Sequence', 'Protein Structure, Secondary', 'Protein Binding', 'Kinetics', 'Mutation', 'Thermodynamics', 'Crystallography, X-Ray', 'Escherichia coli', 'Protein Stability', 'Mice', 'Protein Multimerization', 'Molecular Dynamics Simulation', 'Binding Sites', 'Protein Denaturation', 'Endoplasmic Reticulum', 'Amino Acid Substitution', 'Hydrophobic and Hydrophilic Interactions', 'Protein Transport', 'Hydrogen-Ion Concentration', 'Temperature', 'Catalytic Domain', 'Sequence Homology, Amino Acid', 'Saccharomyces cerevisiae', 'HEK293 Cells', 'Amyloid'
Target	'Kinetics', 'Protein Folding', 'Molecular Dynamics Simulation', 'Protein Conformation', 'Hydrogen-Ion Concentration', 'Thermodynamics'
AttentionXML	'Protein Folding'(✔), 'Molecular Dynamics Simulation'(✔), 'Protein Conformation'(✔), 'Protein Structure, Secondary'(✘), 'Amino Acid Sequence'(✘)
BertXML	'Protein Folding'(✔), 'Protein Structure, Secondary'(✘), 'Protein Structure, Tertiary'(✘), 'Amino Acid Sequence'(✘), 'Protein Conformation'(✔)
LabelCoRank	'Protein Folding'(✔), 'Protein Conformation'(✔), 'Molecular Dynamics Simulation'(✔), 'Kinetics'(✔), 'Thermodynamics'(✔)

LabelCoRank is capable of predicting labels associated with features that are difficult to extract from the text. For instance, in Case 1(Table 10), the text frequently contains high-frequency keywords such as "protein", "fold", and "folding". Thus, other methods can capture the corresponding features and infer keywords like "Protein Folding" and "Protein Conformation". However, the correct labels include terms like "Molecular Dynamics Simulation", "Kinetics", and "Thermodynamics", which are scarcely expressed in the text and difficult to capture based solely on the textual content. Our model expands label information beyond the text by using labels predicted in the first pass combined with a frequency relevance matrix. This allows us to obtain labels like "Kinetics" and "Thermodynamics" in the extended tags. After integrating label embeddings and positional information, the attention mechanism mines text features and incorporates the expanded label features into the final feature expression, enabling the prediction of labels that are difficult to capture from the text alone.

LabelCoRank enhances text feature capture specific to labels. During the process of label sequence expansion, it is not necessary to directly obtain the correct labels; acquiring semantically similar labels also aids in label prediction. For example, in Case 2(Table 11), some keywords appearing in the text do not directly associate

Table 11. Case 2.Extended labels represent additional label information obtained through the correlation matrix. This can be used to analyze why the introduction of additional label information is effective.The prediction results are the top-5 labels predicted by each method.Green: Correct predictions. Red: Incorrect predictions. Orange:Semantics in the text that are directly related to labels.

Text	the tumor suppressor apc differentially regulates multiple β catenin through the function of axin and cki α during c elegans asymmetric stem cell divisions abstract the apc tumor suppressor regulates diverse stem cell processes including regulation of gene expression through wnt β catenin signaling and chromosome stability through microtubule interactions but how the disparate functions of apc are controlled is not well understood acting as part of a wnt signaling pathway β catenin pathway that controls asymmetric cell division caenorhabditis elegans apc apr 1 promotes asymmetric nuclear export signal of the β catenin wrm 1 by asymmetrically stabilizing microtubule wnt signaling pathway function also dependson a second β catenin sys 1 which binds to the c elegans tcf pop 1 to activate gene expression here we show that apr 1 regulates sys 1 levels in asymmetric stem cell orangedivision in addition to its known role in lowering nuclear levels of wrm 1 we demonstrate that sys 1 is also negatively regulated by the c elegans homology of casein kinase 1 1 α ckiα kin 19 we show that kin 19 restricts apr 1 localization thereby regulating nuclear wrm 1 finally the polarity of apr 1 cortical localization is controlled by pry 1 c elegans axin such that pry 1 controls the polarity of both sys 1 and wrm 1 asymmetries we propose a model whereby wnt signaling pathway through ckiα regulates the function of two distinct pools of apc one apc pool negatively regulates sys 1 whereas the second pool stabilizes microtubule and promotes wrm 1 nuclear export signal
Extend labels	'Mice', 'Humans', 'Mice, Inbred C57BL', 'Rats', 'Signal Transduction', 'Mice, Knockout', 'Disease Models, Animal', 'Cells, Cultured', 'Cell Line, Tumor', 'Cell Line', 'Neurons', 'Mice, Transgenic', 'Rats, Sprague-Dawley', 'Cell Proliferation', 'Gene Expression Regulation', 'Mutation', 'Cell Differentiation', 'Protein Binding', 'Mice, Inbred BALB C', 'Phosphorylation', 'Time Factors', 'Brain', 'Amino Acid Sequence', 'HEK293 Cells', 'Cell Movement', 'Rats, Wistar', 'Immunohistochemistry','Macrophages', 'Liver'
Target	'Protein Kinases', 'Adenomatous Polyposis Coli Protein', 'Animals', 'Protein Transport', 'Microtubules', 'Stem Cells', 'Asymmetric Cell Division', 'Cell Nucleus', 'Active Transport, Cell Nucleus', 'Wnt Signaling Pathway', 'Cell Polarity'
AttentionXML	'Animals'(✔), 'Axin Protein(✘)', 'Microtubules'(✔), 'Cell Division'(✘), 'Cell Nucleus'(✔)
BertXML	'Animals'(✔), 'Protein-Serine-Threonine Kinases'(✘), 'Mutation'(✘), 'RNA Interference'(✘), 'Gene Expression Regulation, Developmental'(✘)
LabelCoRank	'Animals'(✔), 'Asymmetric Cell Division'(✔), 'Adenomatous Polyposis Coli Protein'(✔), 'Protein Transport'(✔), 'Stem Cells'(✔)

with labels but are linked through similar semantics, such as "catenin" and "protein" or "elegans" and "animals". Through label expansion, relevant labels may not be directly acquired, but more semantically related labels are obtained. For example, in predicting labels such as "Cell Nucleus" and "Cell Polarity", similar features of labels such as "Cells, Cultured", "Cell Line, Tumor", "Cell Line", and "Cell Differentiation" are utilized through a label attention mechanism. This allows more features related to "Cell" to be extracted, thus aiding label prediction.

5 Conclusion

This paper introduces LabelCoRank, a model that integrates label information by Label Reranking method. The principal mechanism of this model is to expand the labels related to the text by using label frequency correlation information, which allows for the extraction of more effective features targeted at labels through the operation of a label attention mechanism. Additionally, the integration of label frequency distribution information into the label feature sequence via sorting and positional information helps the model to capture more significant features for classification. We compared our model with current advanced models on three public datasets: MAG-CS, PubMed, and AAPD, demonstrating its effectiveness. The efficacy of the modules was validated through ablation studies,

and case analyses illustrated how the introduction of label information via our method positively influences the prediction results.

Acknowledgments

This work was funded by the Fundamental Research Funds for the Central Universities, National High-End Foreign Experts Program(H20240916) and National Natural Science Foundation of China(52374165, 52304185).

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *ArXiv* abs/1803.01271 (2018). https://api.semanticscholar.org/CorpusID:4747877

[2] Ying-Ying Chang, Wei-Yao Wang, and Wen-Chih Peng. 2024. SeGA: Preference-Aware Self-Contrastive Learning with Prompts for Anomalous User Detection on Twitter. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 1 (Mar. 2024), 30–37. https://doi.org/10.1609/aaai.v38i1.27752

[3] Xiaolong Chen, Jieren Cheng, Jingxin Liu, Wenghang Xu, Shuai Hua, Zhu Tang, and Victor S Sheng. 2022. A survey of multi-label text classification based on deep learning. In *International Conference on Adaptive and Intelligent Systems*. Springer, 443–456.

[4] Yahui Chen. 2015. *Convolutional neural network for sentence classification.* Master's thesis. University of Waterloo.

[5] Chaoqun Cui and Caiyan Jia. 2024. Propagation Tree Is Not Deep: Adaptive Graph Contrastive Learning Approach for Rumor Detection. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 1 (Mar. 2024), 73–81. https://doi.org/10.1609/aaai.v38i1.27757

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Yuxiao Dong, Hao Ma, Zhihong Shen, and Kuansan Wang. 2017. A Century of Science: Globalization of Scientific Collaborations, Citations, and Innovations. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 1437–1446. https://doi.org/10.1145/3097983.3098016

[8] Mohamed Goudjil, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali. 2018. A Novel Active Learning Method Using SVM for Text Classification. *International Journal of Automation and Computing* 15, 3 (01 Jun 2018), 290–298. https://doi.org/10.1007/s11633-015-0912-z

[9] Tiquan Gu, Hui Zhao, Zhenzhen He, Min Li, and Di Ying. 2023. Integrating external knowledge into aspect-based sentiment analysis using graph neural network. *Knowledge-Based Systems* 259 (2023), 110025. https://doi.org/10.1016/j.knosys.2022.110025

[10] Hongcheng Guo, Jian Yang, Jiaheng Liu, Jiaqi Bai, Boyang Wang, Zhoujun Li, Tieqiao Zheng, Bo Zhang, Junran Peng, and Qi Tian. 2024. LogFormer: A Pre-train and Tuning Pipeline for Log Anomaly Detection. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 1 (Mar. 2024), 135–143. https://doi.org/10.1609/aaai.v38i1.27764

[11] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-Transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 1315–1325. https://doi.org/10.18653/v1/N19-1133

[12] Sangwoo Han, Eunseong Choi, Chan Lim, Hyunjung Shim, and Jongwuk Lee. 2022. Long-tail Mixup for Extreme Multi-label Classification. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 3998–4002. https://doi.org/10.1145/3511808.3557632

[13] N Jayalakshmi, V Sangeeta, and Appala Srinuvasu Muttipati. 2023. Taylor Horse Herd Optimized Deep Fuzzy clustering and Laplace based K-nearest neighbor for web page recommendation. *Advances in Engineering Software* 175 (2023), 103351. https://doi.org/10.1016/j.advengsoft.2022.103351

[14] Weili Jiang, Kangneng Zhou, Chenchen Xiong, Guodong Du, Chubin Ou, and Junpeng Zhang. 2023. KSCB: a novel unsupervised method for text sentiment analysis. *Applied Intelligence* 53, 1 (01 Jan 2023), 301–311. https://doi.org/10.1007/s10489-022-03389-4

[15] Anwesha Law and Ashish Ghosh. 2022. Multi-Label Classification Using Binary Tree of Classifiers. *IEEE Transactions on Emerging Topics in Computational Intelligence* 6, 3 (2022), 677–689. https://doi.org/10.1109/TETCI.2021.3075717

[16] Fangfang Li, Puzhen Su, Junwen Duan, and Weidong Xiao. 2023. Towards Better Representations for Multi-Label Text Classification with Multi-granularity Information. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9470–9480. https://doi.org/10.18653/v1/2023.findings-emnlp.635

[17] Irene Li, Aosong Feng, Hao Wu, Tianxiao Li, Toyotaro Suzumura, and Ruihai Dong. 2022. LiGCN: Label-interpretable Graph Convolutional Networks for Multi-label Text Classification. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, Lingfei Wu, Bang Liu, Rada Mihalcea, Jian Pei, Yue Zhang, and Yunyao Li (Eds.). Association for Computational Linguistics, Seattle, Washington, 60–70. https://doi.org/10.18653/v1/2022.dlg4nlp-1.7

[18] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 115–124. https://doi.org/10.1145/3077136.3080834

[19] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*. AAAI Press, 2873–2879.

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019). https://api.semanticscholar.org/CorpusID:198953378

[21] Zhiyong Lu. 2011. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database (Oxford)* 2011, 0 (Jan. 2011), baq036.

[22] Qianwen Ma, Chunyuan Yuan, Wei Zhou, and Songlin Hu. 2021. Label-Specific Dual Graph Neural Network for Multi-Label Text Classification. In *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)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 3855–3864. https://doi.org/10.18653/v1/2021.acl-long.298

[23] Yuqing Mao and Zhiyong Lu. 2017. MeSH Now: automatic MeSH indexing at PubMed scale via learning to rank. *Journal of biomedical semantics* 8 (2017), 1–9.

[24] Khalid Nassiri and Moulay Akhloufi. 2023. Transformer models used for text-based question answering systems. *Applied Intelligence* 53, 9 (01 May 2023), 10602–10635. https://doi.org/10.1007/s10489-022-04052-8

[25] Ankit Pal, Muru Selvakumar, and Malaikannan Sankarasubbu. 2020. Multi-Label Text Classification using Attention-based Graph Neural Network. *ArXiv* abs/2003.11644 (2020). https://api.semanticscholar.org/CorpusID:214667462

[26] Anjuman Prabhat and Vikas Khullar. 2017. Sentiment classification on big data using Naïve bayes and logistic regression. In *2017 International Conference on Computer Communication and Informatics (ICCCI)*. 1–5. https://doi.org/10.1109/ICCCI.2017.8117734

[27] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising. In *the 2018 World Wide Web Conference*.

[28] Martha Roseberry, Bartosz Krawczyk, Youcef Djenouri, and Alberto Cano. 2021. Self-adjusting k nearest neighbors for continual learning from multi-label drifting data streams. *Neurocomputing* 442 (2021), 10–25. https://doi.org/10.1016/j.neucom.2021.02.032

[29] Guoying Sun, Yanan Cheng, Fangzhou Dong, Luhua Wang, Dong Zhao, Zhaoxin Zhang, and Xiaojun Tong. 2024. Multi-Label Text Classification model integrating Label Attention and Historical Attention. *Knowledge-Based Systems* 296 (2024), 111878. https://doi.org/10.1016/j.knosys.2024.111878

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[31] Huy-The Vu, Minh-Tien Nguyen, Van-Chien Nguyen, Minh-Hieu Pham, Van-Quyet Nguyen, and Van-Hau Nguyen. 2023. Label-representative graph convolutional network for multi-label text classification. *Applied Intelligence* 53, 12 (01 Jun 2023), 14759–14774. https://doi.org/10.1007/s10489-022-04106-x

[32] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (02 2020), 396–413. https://doi.org/10.1162/qss_a_00021

[33] Qing Wang, Jia Zhu, Hongji Shu, Kwame Omono Asamoah, Jianyang Shi, and Cong Zhou. 2023. GUDN: A novel guide network with label reinforcement strategy for extreme multi-label text classification. *Journal of King Saud University - Computer and Information Sciences* 35, 4 (2023), 161–171. https://doi.org/10.1016/j.jksuci.2023.03.009

[34] Xindi Wang, Robert Mercer, and Frank Rudzicz. 2022. KenMeSH: Knowledge-enhanced End-to-end Biomedical Text Labelling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 2941–2951. https://doi.org/10.18653/v1/2022.acl-long.210

[35] Yan Wang, Zhixuan Chu, Xin Ouyang, Simeng Wang, Hongyan Hao, Yue Shen, Jinjie Gu, Siqiao Xue, James Zhang, Qing Cui, Longfei Li, Jun Zhou, and Sheng Li. 2024. LLMRG: Improving Recommendations through Large Language Model Reasoning Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 17 (Mar. 2024), 19189–19196. https://doi.org/10.1609/aaai.v38i17.29887

[36] Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2024. Exploring Large Language Model for Graph Data Understanding in Online Job Recommendations. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 8 (Mar. 2024), 9178–9186. https://doi.org/10.1609/aaai.v38i8.28769

[37] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-Specific Document Representation for Multi-Label Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 466–475. https://doi.org/10.18653/v1/D19-1044

[38] Linli Xu, Sijie Teng, Ruoyu Zhao, Junliang Guo, Chi Xiao, Deqiang Jiang, and Bo Ren. 2021. Hierarchical Multi-label Text Classification with Horizontal and Vertical Category Correlations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2459–2468. https://doi.org/10.18653/v1/2021.emnlp-main.190

[39] Guangxu Xun, Kishlay Jha, Jianhui Sun, and Aidong Zhang. 2020. Correlation Networks for Extreme Multi-label Text Classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 1074–1082. https://doi.org/10.1145/3394486.3403151

[40] Guangxu Xun, Kishlay Jha, Ye Yuan, Yaqing Wang, and Aidong Zhang. 2019. MeSHProbeNet: a self-attentive probe net for MeSH indexing. *Bioinformatics* 35, 19 (03 2019), 3794–3802. https://doi.org/10.1093/bioinformatics/btz142

[41] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence Generation Model for Multi-label Classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, Emily M. Bender, Leon Derczynski, and Pierre Isabelle (Eds.). Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3915–3926. https://aclanthology.org/C18-1330

[42] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[43] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.). Association for Computational Linguistics, San Diego, California, 1480–1489. https://doi.org/10.18653/v1/N16-1174

[44] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/9e6a921fbc428b5638b3986e365d4f21-Paper.pdf

[45] Delong Zeng, Enze Zha, Jiayi Kuang, and Ying Shen. 2024. Multi-label text classification based on semantic-sensitive graph convolutional network. *Knowledge-Based Systems* 284 (2024), 111303. https://doi.org/10.1016/j.knosys.2023.111303

[46] Ximing Zhang, Qian-Wen Zhang, Zhao Yan, Ruifang Liu, and Yunbo Cao. 2021. Enhancing Label Correlation Feedback in Multi-Label Text Classification via Multi-Task Learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 1190–1200. https://doi.org/10.18653/v1/2021.findings-acl.101

[47] Yu Zhang, Zhihong Shen, Yuxiao Dong, Kuansan Wang, and Jiawei Han. 2021. MATCH: Metadata-Aware Text Classification in A Large Hierarchy. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3246–3257. https://doi.org/10.1145/3442381.3449979

Received 3 August 2024; revised 17 April 2025; accepted 5 June 2025