Journal of Artificial Intelligence Research 82 (2025) 367-388 Submitted 08/2024; published 01/2025

CPM-based Hierarchical Text Classification

Biqing Zeng ZENGBIQING@SCNU.EDU.CN
Yihao Peng PENGYIHAO@M.SCNU.EDU.CN
School of Artificial Intelligence, South China Normal University,
Foshan 528225 China

Jichen Yang NISONYOUNG@GMAIL.COM
School of Cyber Security, Guangdong Polytechnic University,
Guangzhou 510225 China

Peilin Hong HONGPEILIN@M.SCNU.EDU.CN
Junjie Liang 1059316217@QQ.COM
School of Artificial Intelligence, South China Normal University,
Foshan 528225 China

Abstract

In the field of natural language processing, hierarchical text classification (HTC) has emerged as a critical task for organizing and analyzing large volumes of text data. The previous work of HTC often falls short in fully leveraging the hierarchical structure of labels, resulting in suboptimal performance. In addition, it is difficult to capture nuanced relationships between parent and child classes, leading to inaccurate predictions and insufficient differentiation between sibling classes under the same parent category. This gap underscores the need for approaches that can more effectively integrate and utilize both hierarchical and corpus-specific information to improve HTC performance. To address these issues, Concept-aware Prompt Mechanism (CPM) is proposed for HTC, which leverages concept information embedded within hierarchical labels to enhance the representation of these labels and improve classification accuracy. Specifically, we introduce a concept initialization module that extracts concept features from hierarchical labels and a novel concept prompt template to integrate these features into the classification process. Our experimental results demonstrate that the proposed CPM achieves state-of-the-art performance on two benchmark datasets, improving Micro-F1 and Macro-F1 scores to varying degrees, particularly in datasets with complex label hierarchies.

1. Introduction

Text classification, as a significant task in the field of Natural Language Processing (NLP), aims to categorize textual data into different classes or labels, thereby facilitating information retrieval, data analysis, and decision-making processes. Moreover, the applications of text classification are extensive, spanning areas such as spam filtering, sentiment analysis, news classification, and legal document archiving. However, with the increasing volume of information, traditional text classification methods may face certain challenges, especially when it is necessary to subdivide text into multiple hierarchical categories.

Hierarchical Text Classification (HTC), a subfield of text classification, emphasizes a more granular and layered classification system (Silla & Freitas, 2011). Specifically, in HTC, a single data sample can correspond to multiple labels, and these labels possess a clear hierarchical structure, typically represented as a tree structure (as shown in Figure

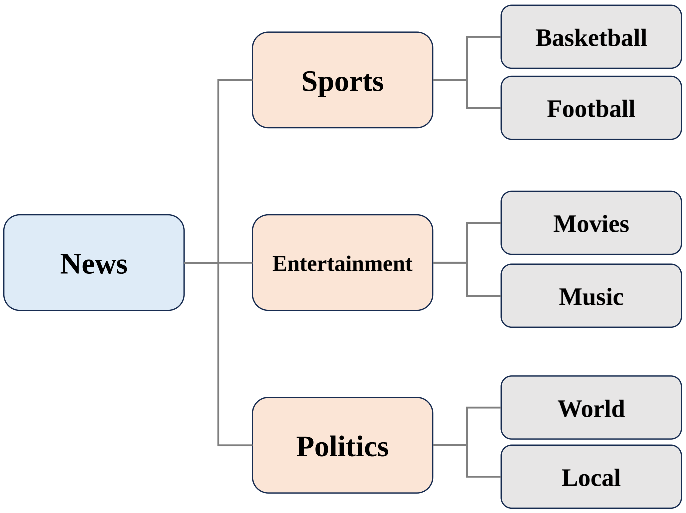

Figure 1: Tree label hierarchy in news hierarchical text classification

1). For instance, an NBA news article can be mapped to both the sports and basketball categories, with basketball evidently being a subclass of sports. Therefore, the importance of HTC lies in its ability to better capture the complex relationships between textual data and the hierarchical structure of labels, thereby providing more accurate and useful classification results. From an application perspective, HTC has been widely utilized in various domains, including news classification (Lewis et al., 2004), advertising systems (Agrawal et al., 2013), information retrieval (Liu et al., 2015), and fine-grained entity typing (Xu & Barbosa, 2018).

Existing HTC models often utilize both text encoder and structure encoder to obtain embeddings for documents and hierarchies, respectively. These embeddings are then integrated through various methods to achieve a composite representation of the text and hierarchical labels (Zhou et al., 2020; Deng et al., 2021; Chen et al., 2021). Although this approach has proven effective, identifying a more suitable method for combining text embeddings and structured label embeddings remains a topic of significant research interest.

In light of this, some scholars have attempted to introduce prompt learning into HTC tasks (Wang et al., 2022b). This approach leverages the flexibility and naturalness of prompt learning, which surpasses traditional fine-tuning in its ability to adapt to a variety of complex text processing tasks, to seamlessly incorporate structured labels into the text. While this method cleverly integrates the hierarchical structure of labels into the prompts, achieving a natural fusion of structural and text embeddings, it fails to account for finer-grained information relationships between parent and child classes. This oversight directly impacts the model's ability to predict classifications accurately, as it does not capture the semantic nuances among sibling classes under the same parent class. Consequently, in some cases, the model struggles to distinguish between similar sibling classes, leading to difficulties in emphasizing different focal points correctly.

Inspired by Wang et al. (2021), we recognize that the integration of conceptual information aids the classifier in better understanding the relationships between parent and child classes as well as among sibling classes if there is a shared concept between subclasses, with a granularity that falls between the subclasses and their parent class.

In this regard, a concept initialization module designed to extract conceptual features embedded within the structured labels is constructed in this work. Wherein, a class's conceptual representation based on the embeddings of high-frequency keywords from documents mapped to the label, along with the embeddings of all its child labels is jointly constructed rather than using a dynamic routing mechanism to derive these concepts. It enables the classifier to more distinctly differentiate the focal points among sibling classes sharing the same parent class, thereby enhancing classification accuracy.

In previous work, HPT (Wang et al., 2022b) has been dedicated to integrating hierarchical information into the prompt template by constructing additional nodes. Inspired by this, we design conceptual nodes for structured labels and inject them into the original label structure. Unlike HPT , the newly generated nodes are not initialized in a random way, because it focuses more on the connection information contained in the tag structure, rather than the simple hierarchical information. Based on this, we use the concept initialization template constructed before to initialize the new node, and then use the method similar to HPT to form the soft prompt template containing the concept information, so as to complete the hierarchical text classification task by using the prompt learning.

The contributions of the work can be summarized as follows:

- Concept-aware Prompt Mechanism (CPM) is proposed for Hierarchical Text Classification (HTC) in this work. To the best of our knowledge, this is the first attempt to combine the conceptual granularity information inherent in hierarchical label structures with prompt learning in the HTC domain.

- A concept initialization module to extract and initialize the implicit concepts within structured labels is proposed. Which enables subsequent structure encoders to explicitly obtain structured label embeddings enriched with these conceptual representations, thereby enhancing the model's understanding of relationships between parent, child, and sibling classes, ultimately improving classification accuracy and hierarchical awareness. We constructed a concept initialization module to extract and initialize the

The rest of the paper is organized as follows: Section 2 introduces related work. The proposed method is introduced in detail in Section 3. Finally, the paper is concluded in Section 4.

2. Related Work

2.1 Hierarchical Text Classification

Existing work on HTC is often categorized based on how they handle label hierarchies into local and global approaches (Wehrmann et al., 2018).

Local approaches construct multiple classifiers according to the hierarchy, combining them into a classifier chain (Cerri et al., 2011). Due to the hierarchical nature inherent in hierarchical multi-label text classification, local approaches typically employ recursive classification methods. This means predicting the top-level labels first, then predicting the next level based on their child labels, and so on, until reaching leaf nodes or final labels. Thus, hierarchical multi-label classification can be seen as a multi-level classification problem, where each level's predictions influence the next. In earlier work, Cerri et al. (2016) proposed using a cascaded multi-layer perceptron model for hierarchical multi-label classification tasks, where each layer of the perceptron corresponds to a layer of the label hierarchy. However, this method, like subsequent local methods, suffers from the problem of exposure bias, where accuracy decreases as the hierarchical depth increases. In recent years, Lok et al. (2023) introduced HJCL, a hierarchical-aware joint supervised contrastive learning method. It uses instance-level and label-level contrastive learning techniques, along with carefully constructed batches, to achieve contrastive learning objectives, indirectly addressing some challenges of local methods such as label hierarchy dependencies and exposure bias. Ma et al. (2023) introduced the Label Correlation Enhanced Decoder (LED), which predicts the presence of class labels in parallel through multi-task learning and captures hierarchical dependencies among labels using hierarchical-aware masks. Wang et al. (2021) enhanced the structural information and relationships among labels by making each layer's classifier obtain further insights through label concepts. In summary, local methods can conveniently extend existing classifiers to hierarchical multi-label classification tasks. However, constructing classifiers this way exposes them to bias, meaning errors in higher-level classifiers propagate to lower levels, causing increasing classification errors as the hierarchy deepens. Current research on local methods aims to provide more information to deep classifiers or reduce errors in shallow classifiers to minimize this bias.

Global approaches treat all categories as a whole, typically constructing only one classifier. Traditional algorithms cannot be directly applied to hierarchical multi-label classification tasks and require redesigning classifiers. Early global methods ignored the hierarchical structure of labels, treating the problem as flat multi-label classification (Johnson & Zhang, 2015). For a long time, simplifying the task to flat multi-label text classification was a mainstream method for addressing HTC issues. However, as research progressed, it became apparent that this approach could lead to label prediction inconsistencies. Recent global method research focuses more on directly encoding the entire label structure through structure encoders to enhance performance. For example, Zhou et al. (2020) designed a structure encoder that integrates prior hierarchical label knowledge to learn label representations. Chen et al. (2020) jointly embedded words and label hierarchies in hyperbolic space. Zhang et al. (2022b) extracted text features according to different hierarchical levels. Deng et al. (2021) introduced information maximization to constrain label representation learning. Zhao et al. (2021) designed an adaptive fusion strategy to extract features from both text and labels. Wang et al. (2022a) proposed a cognitive structure learning model for HTC.

In general, global approaches predominantly rely on the direct encoding of the label hierarchical structure, which represents a notable distinction between CPM and other global methodologies. CPM enhances the model's comprehension of inter-label relationships through conceptual prompts. This divergence confers upon CPM a superior capability in handling texts with intricate hierarchical structures, particularly when shared concepts exist among labels.

2.2 Prompt Tuning

The core idea of prompt-tuning (Schick & Schütze, 2021) is to guide a model to produce more accurate and appropriate outputs by adjusting the prompts given to the model, such as instructions or questions. This method typically involves modifying the model's input to better suit the desired task or context. It can be achieved by altering the input encoding, adding additional contextual information, or adjusting the model's output strategy. In the field of NLP, this approach is frequently used in Masked Language Modeling (MLM) tasks (Devlin et al., 2019).

Specifically, prompt fine-tuning changes the input text or encoding fed into a pretrained language model by constructing and inserting prompt templates. This guides the pretrained language model to more accurately predict the masked words in MLM. During this process, prompts can be presented in various forms, such as hard prompts (Gao et al., 2021) and soft prompts (Qin & Eisner, 2021; Hambardzumyan et al., 2021). Hard prompting is very straightforward, typically involving manually designed fixed templates inserted into the input. For example, manually crafted prompts have been used in sentiment (Zhang et al., 2022a) and medical text (Wang et al., 2023) classification tasks.

In contrast, soft prompting was developed to address the labor-intensive nature of hard prompting. This method either designs algorithms to let the model automatically find some linguistic phrases to form prompts (Shin et al., 2020; Schick et al., 2020) or directly uses learnable vectors as prompts (Liu et al., 2022; Su et al., 2022; Zhu et al., 2024). This allows for more flexibility and efficiency in adapting prompts to the specific requirements of the task, thereby improving the model's performance in various NLP applications.

3. The Proposed Method

In this section, the proposed CPM designed to enhance HTC is introduced in detail. Our method leverages conceptual information derived from hierarchical label structures and integrates it into a pretrained BERT model. This integration is achieved through the construction of virtual nodes initialized with concept representations and their incorporation into soft prompt templates. The following subsections provide a detailed breakdown of the CPM approach, starting with the extraction of concept representations, followed by the construction of the concept prompt template, and finally, the classification process. By systematically embedding conceptual and hierarchical knowledge into the text classification framework, CPM aims to improve the model's understanding and handling of complex label structures. An overview of CPM is given in Figure 2.

3.1 Concept Representation Extraction

In a hierarchical classification structure, concepts serve as granularity information situated between parent and child categories. For a given parent category, its child categories share common "concepts" inherited from the parent. As shown in Figure 3, within the Web of Science (WOS) dataset(Kowsari et al., 2017), the parent label "Medical" encompasses six child category labels: "Diabetes", "Parkinson's Disease", "Alzheimer's Disease", "Anxiety", "Depression", and "Bipolar Disorder." Among these, certain similarities exist between the first three and the last three categories. Specifically, "Diabetes", "Parkinson's Disease" and "Alzheimer's Disease" share the concept of "Chronic disease" while "Anxiety", "Depression"

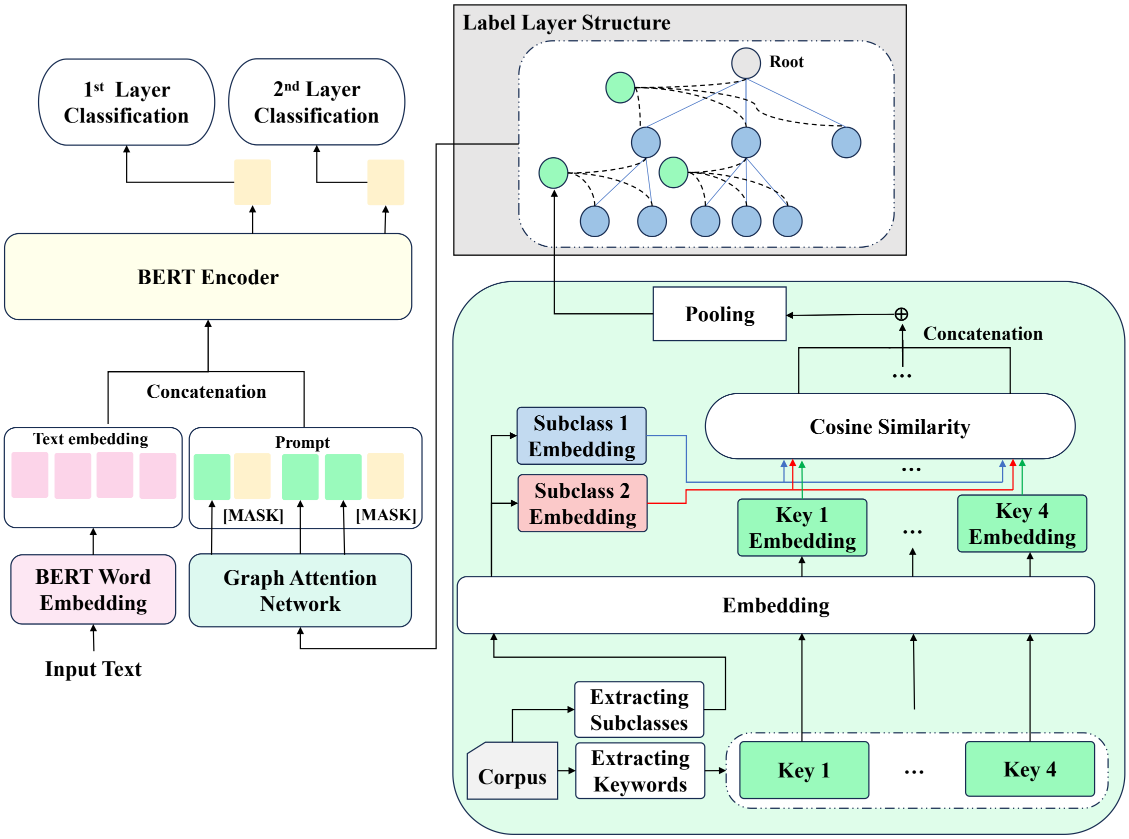

Figure 2: An overview of CPM. CPM integrates conceptual information from label hierarchies into a pretrained BERT model, enhancing its ability to classify texts according to hierarchical structures.

and “Bipolar Disorder” share the concept of “Mental Health”. These concepts occupy an intermediate position between the parent label and the child labels.

In the above example, the concepts of “Chronic disease” and “Mental health” are neither as broad as the parent category “Medical” nor as specific as the child categories like “Diabetes” or “Bipolar Disorder”. Instead, these concepts provide a common theme or characteristic for the related subcategories. Such labels help capture the commonalities and subtle differences between categories, aiding in the understanding and organization of complex hierarchical structures.

To obtain the “concepts” corresponding to a parent node’s category, we extract keywords from the corpus associated with that category and use the top-ranked keywords as the representation for that category. For the WOS, where each document is annotated with several keywords, we simply rank these keywords by their frequency within each category. For datasets without annotated keywords, such as New York Times (NYT) (Sandhaus,

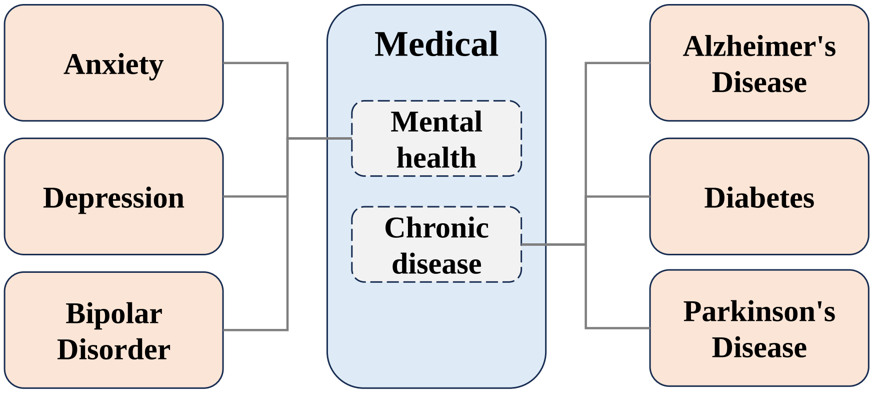

Figure 3: Tree label hierarchy in news hierarchical text classification

2008), we use the TF-IDF algorithm to extract them. In our work, we selected the top 4 keywords.

Once the keywords are obtained, we generate or retrieve their embeddings using a pretrained word embedding model, such as Word2Vec, GloVe, or BERT. In our study, we chose BERT as the pretrained model for obtaining embeddings. For each child category embedding, we calculate its cosine similarity with each concept embedding. Cosine similarity measures the degree of similarity between two vectors based on their direction.Then, for a given keyword, we create a composite representation of each concept by performing a weighted average of the child category embeddings, where the weights are determined by the similarity scores. In this manner, more similar child categories contribute more significantly to the concept representation. This approach allows us to effectively integrate conceptual information into the hierarchical classification model, enhancing its ability to capture relationships and distinctions within the hierarchy.

Formally, for a non-leaf node c in the structured labels of a given dataset, with a set of subclasses $Sub = \{c_1, c_2, \dots, c_n\}$ and 4 keywords $K = \{k_1, \dots, k_4\}$, where n is the number of subclasses of c. For ease of understanding, we take the "Civil" category from the WOS dataset as an example. After extraction, we obtained several of its subclasses along with four keywords, as detailed in Table 1 below. Then, we compute the cosine similarity for each of subclass c_i and keyword k_j as follows:

$$\mathrm{sim}(\boldsymbol{c}_i, \boldsymbol{k}_j) = \frac{\boldsymbol{c}_i \cdot \boldsymbol{k}_j}{\|\boldsymbol{c}_i\| \|\boldsymbol{k}_j\|} \tag{1}$$

where $\boldsymbol{c}_i$ is the embedding of c_i, and $\boldsymbol{k}_j$ is the embedding of k_j, obtained through a BERT encoder. The weight $w_{i,j}$ of each subclass c_i for keyword k_j is determined by the cosine similarity:

$$w_{i,j} = \frac{\mathrm{sim}(\boldsymbol{c}_i, \boldsymbol{k}_j)}{\sum_{k=1}^{n} \mathrm{sim}(\boldsymbol{c}_k, \boldsymbol{k}_j)} \tag{2}$$

Subclasses	Keywords
Water Pollution, Suspension Bridge, Geotextile, Green Building, Smart Material, Stealth Technology, Ambient Intelligence, Solar Energy, Remote Sensing, Construction Management, Rainwater Harvesting	water, management, construction, energy

Table 1: Subclasses and keywords for the Civil category in WOS

Next, we create the representation of each keyword by taking the weighted average of the subclass embeddings:

$$\boldsymbol{R}_j = \sum_{i=1}^{m} w_{i,j} \cdot \boldsymbol{c}_i \tag{3}$$

We then concatenate the representations of all keywords and perform pooling to obtain the concept representation of class c:

$$\boldsymbol{R}^{\text{key}} = [\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_n] \tag{4}$$

$$\boldsymbol{r}^{\text{concept}} = \text{AvgPool}(\boldsymbol{R}^{\text{key}}) \tag{5}$$

Here, AvgPool() denotes the average pooling operation. We repeat the above steps for every non-leaf node to obtain the concept representations for all non-leaf node labels. These concept representations are used to initialize the virtual nodes constructed in the next step.

Specifically, for the topmost nodes in the label structure, since they do not share a common parent, we artificially assume a class named “root” as their parent node and define the concept representation of the root using random initialization.

3.2 Constructing the Concept Prompt Template

As previously mentioned, conceptual information should provide significant support for HTC tasks. To utilize this conceptual information, we need to integrate the concepts into the original label graph structure. In HPT, researchers integrate hierarchical information into virtual nodes by creating one virtual node for each level of the label structure. Inspired by this, we have adopted the approach of creating virtual nodes. However, unlike HPT, which primarily aims for hierarchical awareness, we leverage the concept representations obtained earlier. Therefore, we create a virtual node $\boldsymbol{t}$ for each non-leaf node, initialized with the concept representation derived from equation (5). Node $\boldsymbol{t}$ is not only connected to the non-leaf node but also to all its child nodes.

After obtaining the virtual nodes initialized with concept representations, we input the new graph, which includes these virtual nodes, into a graph encoder to obtain the representations of all virtual nodes. Each virtual node learns hierarchical knowledge integrated with the concept:

$$\boldsymbol{e}_i^{\text{concept}} = \text{GAT}(\boldsymbol{r}_i^{\text{concept}}, G : \boldsymbol{r}_i^{\text{concept}} \in G) \tag{6}$$

where G is the label graph structure with the virtual nodes added, $\boldsymbol{r}_i^{\text{concept}}$ is the initial feature of virtual node i, and $\boldsymbol{e}_i^{\text{concept}}$ is its representation updated by the Graph Attention Network (GAT) (Velickovic et al., 2018). GAT is a type of Graph Neural Network that employs an attention mechanism to assign different weights to nodes in the graph, allowing it to capture the complex relationships between nodes. Essentially, GAT computes attention coefficients between nodes to weight the features of neighboring nodes, which are learned and reflect the importance of these neighbors. This makes GAT particularly suitable for processing graph-structured data, such as our label hierarchy, where the relationships between nodes are crucial for understanding the overall structure.

The representations of the virtual nodes are used as the soft prompt template in this study. For a text x, the soft prompt attaches a fixed number of learnable virtual template words to the text as a template, e.g., "[CLS] x [SEP] [V1] [MASK] [V2] [MASK] ... [Vn] [MASK] [SEP]," where $\{[V1], [V2], \ldots, [Vn]\}$ are virtual template words, n is the number of virtual nodes, and [MASK] are the masked words to be predicted. During training, the pretrained language model learns to predict "[MASK]" and adjust the virtual template words.

In practice, we cannot directly define the virtual template words, but we can obtain the concept representations corresponding to these virtual template words from equation (5). Thus, we can concatenate the embeddings of the original text with the concept representations, indirectly achieving the concatenation of the prompt template with the original text:

$$T = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{e}_1^{\text{concept}}, \boldsymbol{e}_{\text{MASK}}, \boldsymbol{e}_2^{\text{concept}}, \ldots, \boldsymbol{e}_{M+1}^{\text{concept}}, \boldsymbol{e}_{\text{MASK}}, \ldots] \tag{7}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ are the word embeddings of the original text, M is the number of shared concepts in the second layer of the label structure, which is equal to the number of parent nodes in that layer. For simplicity, equation (7) only denotes the concept representations shared by the first and second layers. $\boldsymbol{e}_{\text{MASK}}$ is the embedding of [MASK], initialized by the BERT [MASK] token.

Then, we encode T through the BERT encoding layer to obtain their hidden states:

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N, \boldsymbol{h}_1^{\text{concept}}, \boldsymbol{h}_{\text{MASK}}, \boldsymbol{h}_2^{\text{concept}}, \ldots, \boldsymbol{h}_{M+1}^{\text{concept}}, \boldsymbol{h}_{\text{MASK}}, \ldots] \tag{8}$$

where $\boldsymbol{h}_i^{\text{concept}}$ is the hidden state of the i-th $\boldsymbol{r}_{\text{concept}}$, corresponding to the i-th layer of the label hierarchy.

3.3 Classification

As in the previous work, we restrict each [MASK] token to predict only the labels at its respective hierarchical level. Specifically, during the prediction phase, we compare the hidden state vector $\boldsymbol{h}_{\text{MASK}}^i$ at the [MASK] position with the label embedding vectors $\boldsymbol{e}_j^i$ of the corresponding layer in the label structure. The score vector $\boldsymbol{s}$ is calculated through a dot product operation, where the elements s_j of the score vector are computed as follows:

$$s_j = \boldsymbol{h}_{\text{MASK}}^i \cdot \boldsymbol{e}_j^i \tag{9}$$

Here, $\boldsymbol{h}^i_{\text{MASK}}$ is the hidden state vector at the [MASK] position in layer i, $\boldsymbol{e}^i_j$ is the embedding vector of the j-th label in layer i, and n is the dimension of the vectors.

Each element s_j in the score vector $\boldsymbol{s}$ represents the degree of match for the j-th label word in the given context at that layer. The word corresponding to the highest-scoring element in the score vector $\boldsymbol{s}$ is the model's predicted [MASK] word. This indicates that the model believes this word best fits the [MASK] position in the given context. The classification process is shown in Figure 4.

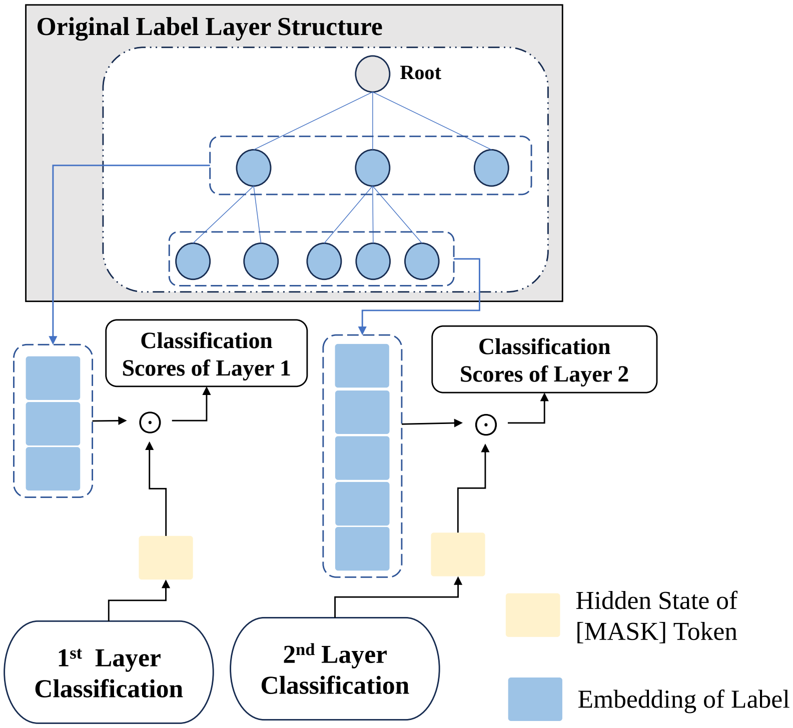
Original Label Layer Structure
Root
Classification Scores of Layer 1
Classification Scores of Layer 2
⊙
⊙
1st Layer Classification
2nd Layer Classification
Hidden State of [MASK] Token
Embedding of Label

Figure 4: The process of the classification mechanism using hidden state vectors and label embeddings. The score vectors are computed using a dot product operation to predict the best fitting label words for the [MASK] position.

4. Evaluation and Analysis

4.1 Datasets

To comprehensively evaluate the performance of our model, we conducted experiments and analysis on two datasets: Web-of-Science (WOS) and New York Times (NYT). Both datasets are publicly available, with their original versions obtainable from the following sources: the WOS dataset can be accessed at `https://github.com/kk7nc/HDLTex`, and the NYT dataset is available at `https://catalog.ldc.upenn.edu/LDC2008T19`. The reason why we selected them is that both of them have diverse label structures, which allow us to test the model's ability to handle both simple and complex hierarchical relationships. In addition, the WOS dataset contains abstracts of published papers from the Web of Science, while the NYT dataset is used for news classification. Some details about WOS and NYT are given in Table 2.

Dataset	Depth level number	Train (#)	Val (#)	Test (#)	Label number
WOS	2	30070	7518	9397	141
NYT	8	23345	5834	7292	166

Table 2: Some details about WOS and NYT datasets. In which, depth level indicates the maximum number of levels in the hierarchical label structure, while label number denotes the total count of unique labels within the dataset, # stands for the number of document.

From Table 2, it can be seen that NYT has more depth level number than WOS, in other words, NYT has more complex structure than that of WOS.

4.2 Evaluation Metrics

To facilitate comparison with other HTC works, we adopted the most commonly used evaluation metrics, Micro-F1 and Macro-F1 scores, as standard evaluation metrics (Sun et al., 2024; Yan et al., 2024). Micro-F1 score is calculated by aggregating the total true positives (TP), false positives (FP), and false negatives (FN) across all labels to compute a global average F1 score, reflecting the model's overall performance on all labels. However, this metric may favor frequently occurring labels, making it less effective in accurately reflecting the model's performance on less frequent labels. On the other hand, Macro-F1 score calculates the F1 score for each label and averages them, giving equal importance to all labels, which is suitable for handling label imbalance issues. These two metrics help in comprehensively evaluating the model's performance across different tasks and datasets. Their calculations are as follows:

$$\text{Micro-F1} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{10}$$

$$\text{Macro-F1} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{2 \times \text{TP}_i}{2 \times \text{TP}_i + \text{FP}_i + \text{FN}_i} \right) \tag{11}$$

where TP (True Positive), FP (False Positive), and FN (False Negative) represent the number of correct classifications, incorrect classifications, and missed classifications made by the model, respectively. In the calculation of Macro-F1, TP_i, FP_i and FN_i denote the true positives, false positives, and false negatives for the i-th label. n represents the total number of labels in the dataset. By averaging the F1 scores of each label, Macro-F1 gives equal importance to all labels.

4.3 Implementation Details

We implemented our model in an end-to-end manner using PyTorch and conducted a series of experiments on an NVIDIA 3090 GPU. Following the approach of previous work (HPT), we used bert-base-uncased as our base architecture. The batch size was set to 16, the optimizer was Adam, and the learning rate was 3e-5. Typically, for each parent class, we selected the top 4 extracted keywords to generate concepts. Unless we were specifically evaluating the impact of the number of selected keywords, we did not adjust any other hyperparameters in other experiments. During training, we set the stopping criteria to halt the training process if both Macro-F1 and Micro-F1 scores did not improve over 5 epochs.

4.4 Experimental Results and Analysis

The experimental results across different layers and the overall results on the WOS and NYT datasets in terms of Micro-F1 and Macro-F1 are presented in Table 3.

Dataset	Layer	Micro-F1 (%)	Macro-F1 (%)
WOS	Layer 1	90.41	89.03
	Layer 2	84.28	75.59
	Overall	87.33	81.96
NYT	Layer 1	90.77	89.97
	Layer 2	83.01	77.58
	Layer 3	76.57	73.21
	Layer 4	74.93	69.42
	Layer 5	76.01	71.06
	Layer 6	74.41	68.95
	Layer 7	73.86	62.68
	Layer 8	71.87	62.02
	Overall	81.54	72.12

Table 3: The experimental results of CPM for each layer and the overall results on the WOS and NYT datasets in terms of Micro-F1 and Macro-F1, respectively.

From Table 3, several conclusions can be drawn:

- The proposed method can give good performance across various hierarchical levels, which demonstrates the proposed CPM has strong classification ability in both simpler hierarchical structures such as WOS and more complex ones such as NYT.
- On the WOS dataset, CPM achieves a high Micro-F1 score of 90.41% and a Macro-F1 score of 89.03% at the first layer, indicating the model's capability to handle top-level categories with high precision and recall. However, as we move to the second layer, there is a noticeable drop in both Micro-F1 and Macro-F1 scores, which fall to 84.28% and 75.59%, respectively. The reason may be that the classification task becomes more challenging with more specific categories. The overall scores for WOS are 87.33% and 81.96%, reflecting robust overall performance across the hierarchical structure.

- In contrast, the performance on the NYT dataset reveals a more complex scenario. At the first layer, the performance is comparable to WOS, with Micro-F1 and Macro-F1 scores of 90.77% and 89.97%, respectively, suggesting that the model effectively distinguishes high-level categories in both datasets. However, there is a more pronounced decline in performance with increasing layers in the NYT dataset, especially beyond the third layer. For instance, at layer 3, the scores drop to 76.57% and 73.21% and further decrease at deeper layers, such as layer 8, with scores of 71.87% and 62.02%. The overall scores for NYT are 81.54% and 72.12%, respectively. These lower scores compared to WOS are attributed to the more complex hierarchical structure of the NYT dataset, which presents a greater challenge for hierarchical classification.

- The results also reveal the exposure bias issue, where accuracy tends to decrease as the hierarchical depth increases. This is evident in both datasets but is more pronounced in NYT due to its deeper and more intricate hierarchy. Despite this, the method's ability to maintain relatively high scores even at deeper layers, such as layers 5 and 6 in NYT, indicates its effectiveness in capturing hierarchical dependencies.

- In summary, the experimental results affirm the efficacy of our proposed method across various hierarchical levels, showcasing its capability to adapt to both simple and complex hierarchical structures.

4.5 Performance Comparison

During the process of model performance evaluation, we conducted various experiments to comprehensively understand the characteristics of the model, including ablation experiments and experiments on the impact of the number of keywords on model performance, which will be elaborated in detail below.

The main distinction between our work and previous works lies in injecting implicit conceptual information from the label hierarchy into virtual nodes to form prompts with more integrated label structure information. To illustrate the effectiveness of conceptual information and virtual nodes, ablation study is performed and the corresponding results on WOS and NYT in terms of Micro-F1 and Macro-F1 are given in Table 4. Wherein the first line and the second line in Table 4 are about concept and concept plus virtual node, respectively, which can be used to observe the performance of the modules of conceptual information and virtual nodes.

Method	WOS		NYT	
	Micro-F1 (%)	Macro-F1 (%)	Micro-F1 (%)	Macro-F1 (%)
w/o concept	86.98	81.12	80.38	70.51
w/o concept & virtual node	85.21	79.90	78.21	69.02
Ours	**87.33**	**81.96**	**81.54**	**72.12**

Table 4: The ablation study results about the modules of conceptual information and virtual nodes on WOS and NYT in terms of Micro-F1 and Macro-F1. In which, the bold values indicate the best-performing results for each metric on both datasets.

According to Table 4, when the concept module is removed (w/o concept), both Micro-F1 and Macro-F1 metrics decrease, with a more pronounced decline on the NYT dataset, which has deeper and more complex label structures. This demonstrates the importance of conceptual information in complex label structures. When virtual nodes are further removed (w/o concept & virtual node), both Micro-F1 and Macro-F1 further decrease, indicating that virtual nodes play a crucial role in deep label structures by enhancing connections and feature sharing among sibling nodes, significantly improving classification performance.

In addition to the ablation experiment, we also investigated the influence of the number of keywords on the model performance.

Regarding the selection of the number of keywords, we tested our model on both datasets within the range of 1 to 10 keywords. The experimental results are shown in Figure 5.

From Figure 5, it can be found that the two evaluation metrics on both datasets show a trend of first rising and then falling. This demonstrates that the number of selected keywords has a significant impact on the model performance. With a moderate number of keywords (e.g., 4), the model can fully utilize conceptual information to improve Micro-F1 and Macro-F1 metrics. In both WOS and NYT datasets, 4 keywords showed the best performance, indicating that 4 keywords provide sufficient conceptual information without introducing too much redundancy and noise. When the number of keywords exceeds 4, the model's performance slightly declines, possibly due to the introduction of redundant information, which adversely affects classification performance.

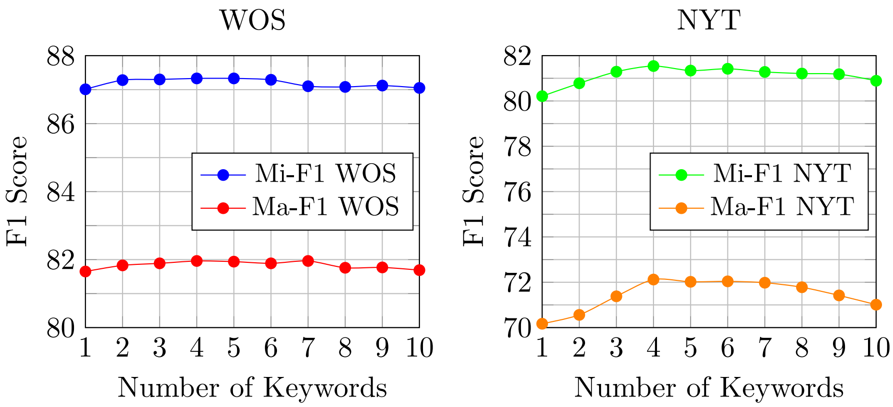

Figure 5: Line charts of performance metrics for different numbers of keywords on WOS and NYT datasets.

4.6 Comparison with Some Known Systems

Table 5 gave the experimental results on WOS and NTT in terms of Micro-F1 and Macro-F1 among the proposed method and some known systems.

- **HiLAP** (Mao et al., 2019): This method formulates the HTC problem as a Markov decision process and proposes a label assignment strategy learned through deep reinforcement learning. It determines the placement of objects and when to stop the assignment process.

- **HiAGM** (Zhou et al., 2020): Utilizing a structure encoder, HiAGM models label dependencies both top-down and bottom-up based on prior hierarchical knowledge. This enables a comprehensive understanding and efficient classification of hierarchical label structures.
- **BERT** (Devlin et al., 2019): BERT is a widely used pretrained model that employs bidirectional encoder representations from massive textual data. It captures deep semantic relationships through masked language modeling and next sentence prediction tasks, achieving significant performance improvements across various natural language processing tasks.
- **HPT** (Wang et al., 2022b): This method introduces a hierarchical prompt tuning approach by constructing dynamic virtual templates. It integrates label hierarchy knowledge into a multi-label masked language model and introduces a zero-boundary multi-label cross-entropy loss to align HTC and MLM task objectives.
- **HGCLR** (Wang et al., 2022a): It embeds the label hierarchy structure directly into the text encoder for modeling. During training, HGCLR constructs positive samples for the input text under the guidance of the label hierarchy. By bringing the input text closer to its positive samples, the text encoder independently generates hierarchy-aware text representations.
- **LSE+HiAGM** (Sun et al., 2024): This approach defines a common density coefficient to measure the importance of label pairs and uses this coefficient to update topological structure features, enabling global label association. By integrating topological structure features, textual features, and hierarchical label features, it improves the embedding quality of lower-level labels.
- **Seq2Label** (Yan et al., 2024): It utilizes a label sequence random shuffling mechanism to map text to multiple different sequences of labels during training. This avoids the limitations of using only one specific sequence order.

From Table 5, several conclusion can be drawn:

- On the WOS, the proposed CPM performs a little better than the traditional methods. For instance, the Micro-F1 can be increased 0.17% and 0.02% compared with HPT (87.16%) and Seq2Label (87.31%), respectively. This may be due to the WOS dataset's label structure having only 2 levels, where the conceptual information within the label hierarchy is relatively simple. Therefore, our concept-based prompt method does not significantly improve over the existing best models in this case.
- On the NYT, the CPM performs much better than the traditional methods. For example, the Micro-F1 can be improved 1.12% and 2.68% compared with HPT and HGCLR, respectively. It indicates better classification capability of CPM for global labels. The significant increase in Macro-F1 (72.12%) further validates our method's

Model	WOS		NYT	
	Micro-F1 (%)	Macro-F1 (%)	Micro-F1 (%)	Macro-F1 (%)
HiLAP	82.76	59.94	73.52	51.47
HiAGM	85.82	80.28	74.97	60.83
BERT+HiAGM	86.04	80.19	78.64	66.76
HGCLR	87.11	81.20	78.86	67.96
HPT	87.16	81.93	80.42	70.42
LSE+HiAGM	86.01	80.01	75.01	61.29
Seq2Label	87.31	81.86	-	-
Ours	**87.33**	**81.96**	**81.54**	**72.12**

Table 5: Experimental results comparison among the proposed systems with the state-of-the-art systems on WOS and NYT in terms of Micro-F1 and Macro-F1. The experimental data for these systems all come from their original papers.

advantage in handling label imbalance issues. The NYT dataset's label structure reaches a maximum depth of 8 levels, making the hierarchical concepts more complex and rich. Therefore, our concept-based prompt method shows more significant performance improvement on this dataset.

- From the perspective of the graph encoder, the introduction of virtual nodes potentially enhances the connections between sibling nodes. In traditional GAT models, node feature updates primarily rely on features from directly connected neighbors, and the influence of indirectly connected nodes is relatively small. By introducing virtual nodes, these sibling nodes can be indirectly connected and share features through the virtual nodes, enhancing their interconnections. This enhanced connection captures richer hierarchical information. In the WOS dataset, although our model shows only minor improvements in Micro-F1 and Macro-F1, it indicates that the hierarchical awareness enhancement is limited in simpler label structures. However, this approach demonstrates significant advantages in datasets with complex label structures, such as NYT.

- Compared to HPT, the increase in the number of virtual nodes might be one reason for our method's superior performance. In HPT, only one virtual node is added for each layer in the label hierarchy, whereas our method adds a virtual node for each non-leaf node. As the depth of the label hierarchy increases, the number of non-leaf nodes often increases significantly, and accordingly, the number of virtual nodes we construct also increases. This extensive introduction of virtual nodes enhances the propagation of hierarchical information, resulting in better performance on datasets with deeper label structures.

- Our comparative analysis encompasses a variety of models that represent the spectrum of approaches to hierarchical text classification. These models range from established methods in reinforcement learning to modern techniques involving structural encoding and prompt tuning. The selection spans different years, illustrating the progression of methodologies in the field. This comparison not only benchmarks our proposed CPM against both classical and contemporary approaches but also highlights its consistent performance advantages, regardless of the evolving nature of NLP research and applications.

5. Conclusion

In this paper, we introduced CPM, an innovative concept-aware prompt mechanism aimed at improving hierarchical text classification. By extracting and utilizing granular concept-level information from hierarchical label structures, we enhance the understanding of parent-child and sibling relationships. Our approach involves constructing virtual nodes initialized with concept representations and integrating them into soft prompt templates to guide the classification process.

Our experimental results on the WOS and NYT datasets highlight the effectiveness of CPM. The method achieves improvements in Micro-F1 and Macro-F1 scores, particularly on the NYT dataset, which has a deeper and more complex label hierarchy. This indicates that our approach has advantages in capturing complex hierarchical relationships and handling label imbalances. However, we also observed that CPM's improvements are more limited when the label hierarchy is shallow, as seen with the WOS dataset. This suggests that in simpler label structures, the hierarchical awareness enhancement is less pronounced. This is because the conceptual information embedded in such label structures is relatively simple, and the concept-based prompting method does not significantly outperform existing best models.

Future work will focus on the following aspects to improve CPM's performance on datasets with simpler label structures.

- Further refining the concept initialization module to better adapt to varying depths of label hierarchies, thereby enhancing performance on datasets with shallow label hierarchies. In addition, CPM can dynamically adjust to different levels of hierarchical complexity by improving the prompt template design.
- Further optimization and extension of CPM have the potential to showcase robust performance across a broader range of applications.

Acknowledgments

We would like to extend our sincere gratitude to all collaborators who have participated in this study. We also thank our anonymous reviewers for their comments. Both Biqing Zeng and Jichen Yang are corresponding authors.

References

Agrawal, R., Gupta, A., Prabhu, Y., & Varma, M. (2013). Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In *22nd International*

World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, pp. 13–24. International World Wide Web Conferences Steering Committee / ACM.

Cerri, R., Barros, R. C., de Carvalho, A. C. P. L. F., & Jin, Y. (2016). Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinform.*, *17*, 373.

Cerri, R., Barros, R. C., & de Leon Ferreira de Carvalho, A. C. P. (2011). Hierarchical multi-label classification for protein function prediction: A local approach based on neural networks. In Ventura, S., Abraham, A., Cios, K. J., Romero, C., Marcelloni, F., Benítez, J. M., & Galindo, E. L. G. (Eds.), *11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22-24, 2011*, pp. 337–343. IEEE.

Chen, B., Huang, X., Xiao, L., Cai, Z., & Jing, L. (2020). Hyperbolic interaction model for hierarchical multi-label classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7496–7503. AAAI Press.

Chen, H., Ma, Q., Lin, Z., & Yan, J. (2021). Hierarchy-aware label semantics matching network for hierarchical text classification. In Zong, C., Xia, F., Li, W., & Navigli, R. (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 4370–4379. Association for Computational Linguistics.

Deng, Z., Peng, H., He, D., Li, J., & Yu, P. S. (2021). Htcinfomax: A global model for hierarchical text classification via information maximization. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., & Zhou, Y. (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 3259–3265. Association for Computational Linguistics.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., & Solorio, T. (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics.

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. In Zong, C., Xia, F., Li, W., & Navigli, R. (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 3816–3830. Association for Computational Linguistics.

Hambardzumyan, K., Khachatrian, H., & May, J. (2021). WARP: word-level adversarial reprogramming. In Zong, C., Xia, F., Li, W., & Navigli, R. (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 4921–4933. Association for Computational Linguistics.

Johnson, R., & Zhang, T. (2015). Effective use of word order for text categorization with convolutional neural networks. In Mihalcea, R., Chai, J. Y., & Sarkar, A. (Eds.), *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pp. 103–112. The Association for Computational Linguistics.

Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017). Hdltex: Hierarchical deep learning for text classification. In Chen, X., Luo, B., Luo, F., Palade, V., & Wani, M. A. (Eds.), *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*, pp. 364–371. IEEE.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, *5*, 361–397.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., & Tang, J. (2022). P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Muresan, S., Nakov, P., & Villavicencio, A. (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 61–68. Association for Computational Linguistics.

Liu, X., Gao, J., He, X., Deng, L., Duh, K., & Wang, Y. (2015). Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In Mihalcea, R., Chai, J. Y., & Sarkar, A. (Eds.), *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pp. 912–921. The Association for Computational Linguistics.

Lok, S. C., He, J., Gutiérrez-Basulto, V., & Pan, J. Z. (2023). Instances and labels: Hierarchy-aware joint supervised contrastive learning for hierarchical multi-label text classification. In Bouamor, H., Pino, J., & Bali, K. (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 8858–8875. Association for Computational Linguistics.

Ma, K., Huang, Z., Deng, X., Guo, J., & Qiu, W. (2023). LED: label correlation enhanced decoder for multi-label text classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pp. 1–5. IEEE.

Mao, Y., Tian, J., Han, J., & Ren, X. (2019). Hierarchical text classification with reinforced label assignment. In Inui, K., Jiang, J., Ng, V., & Wan, X. (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and* *the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 445–455. Association for Computational Linguistics.

Qin, G., & Eisner, J. (2021). Learning how to ask: Querying lms with mixtures of soft prompts. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., & Zhou, Y. (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 5203–5212. Association for Computational Linguistics.

Sandhaus, E. (2008). The new york times annotated corpus. Linguistic Data Consortium, Philadelphia. 6(12):e26752.

Schick, T., Schmid, H., & Schütze, H. (2020). Automatically identifying words that can serve as labels for few-shot text classification. In Scott, D., Bel, N., & Zong, C. (Eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 5569–5578. International Committee on Computational Linguistics.

Schick, T., & Schütze, H. (2021). Exploiting cloze-questions for few-shot text classification and natural language inference. In Merlo, P., Tiedemann, J., & Tsarfaty, R. (Eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pp. 255–269. Association for Computational Linguistics.

Shin, T., Razeghi, Y., IV, R. L. L., Wallace, E., & Singh, S. (2020). Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In Webber, B., Cohn, T., He, Y., & Liu, Y. (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 4222–4235. Association for Computational Linguistics.

Silla, C. N., & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov.*, *22*(1-2), 31–72.

Su, Y., Wang, X., Qin, Y., Chan, C., Lin, Y., Wang, H., Wen, K., Liu, Z., Li, P., Li, J., Hou, L., Sun, M., & Zhou, J. (2022). On transferability of prompt tuning for natural language processing. In Carpuat, M., de Marneffe, M., & Ruíz, I. V. M. (Eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 3949–3969. Association for Computational Linguistics.

Sun, H., He, X., & Peng, Y. (2024). HCL: hierarchical consistency learning for webly supervised fine-grained recognition. *IEEE Trans. Multim.*, *26*, 5108–5119.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Wang, X., Zhao, L., Liu, B., Chen, T., Zhang, F., & Wang, D. (2021). Concept-based label embedding via dynamic routing for hierarchical text classification. In Zong, C., Xia, F., Li, W., & Navigli, R. (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 5010–5019. Association for Computational Linguistics.

Wang, Y., Wang, Y., Peng, Z., Zhang, F., Zhou, L., & Yang, F. (2023). Medical text classification based on the discriminative pre-training model and prompt tuning. *Digital Health*, *9*, 20552076231193213. Art. no. 20552076231193213.

Wang, Z., Wang, P., Huang, L., Sun, X., & Wang, H. (2022a). Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In Muresan, S., Nakov, P., & Villavicencio, A. (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 7109–7119. Association for Computational Linguistics.

Wang, Z., Wang, P., Liu, T., Lin, B., Cao, Y., Sui, Z., & Wang, H. (2022b). HPT: hierarchy-aware prompt tuning for hierarchical text classification. In Goldberg, Y., Kozareva, Z., & Zhang, Y. (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 3740–3751. Association for Computational Linguistics.

Wehrmann, J., Cerri, R., & Barros, R. C. (2018). Hierarchical multi-label classification networks. In Dy, J. G., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 5225–5234. PMLR.

Xu, P., & Barbosa, D. (2018). Neural fine-grained entity type classification with hierarchy-aware loss. In Walker, M. A., Ji, H., & Stent, A. (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 16–25. Association for Computational Linguistics.

Yan, J., Li, P., Chen, H., Zheng, J., & Ma, Q. (2024). Does the order matter? A random generative way to learn label hierarchy for hierarchical text classification. *IEEE ACM Trans. Audio Speech Lang. Process.*, *32*, 276–285.

Zhang, H., Zhang, X., Huang, H., & Yu, L. (2022a). Prompt-based meta-learning for few-shot text classification. In Goldberg, Y., Kozareva, Z., & Zhang, Y. (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 1342–1357. Association for Computational Linguistics.

Zhang, X., Xu, J., Soh, C., & Chen, L. (2022b). LA-HCN: label-based attention for hierarchical multi-label text classification neural network. *Expert Syst. Appl.*, *187*, 115922.

Zhao, R., Wei, X., Ding, C., & Chen, Y. (2021). Hierarchical multi-label text classification: Self-adaption semantic awareness network integrating text topic and label level information. In Qiu, H., Zhang, C., Fei, Z., Qiu, M., & Kung, S. (Eds.), *Knowledge Science, Engineering and Management - 14th International Conference, KSEM 2021, Tokyo, Japan, August 14-16, 2021, Proceedings, Part II*, Vol. 12816 of *Lecture Notes in Computer Science*, pp. 406–418. Springer.

Zhou, J., Ma, C., Long, D., Xu, G., Ding, N., Zhang, H., Xie, P., & Liu, G. (2020). Hierarchy-aware global model for hierarchical text classification. In Jurafsky, D., Chai, J., Schluter, N., & Tetreault, J. R. (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 1106–1117. Association for Computational Linguistics.

Zhu, Y., Wang, Y., Mu, J., Li, Y., Qiang, J., Yuan, Y., & Wu, X. (2024). Short text classification with soft knowledgeable prompt-tuning. *Expert Syst. Appl.*, *246*, 123248.