Exploiting Global and Local Hierarchies for Hierarchical Text Classification

Ting Jiang[1], **Deqing Wang**[1,4,*], **Leilei Sun**[1],
Zhongzhi Chen[2], **Fuzhen Zhuang**[1,3,4], **Qinghong Yang**[2]
[1]SKLSDE, School of Computer, Beihang University, Beijing, China
[2]School of Software, Beihang University, Beijing, China
[3]Institute of Artificial Intelligence, Beihang University, Beijing, China
[4]Zhongguancun Laboratory, Beijing, China
{royokong, dqwang, leileisun, jongjyh, zhuangfuzhen, yangqh}@buaa.edu.cn

Abstract

Hierarchical text classification aims to leverage label hierarchy in multi-label text classification. Existing methods encode label hierarchy in a global view, where label hierarchy is treated as the static hierarchical structure containing all labels. Since global hierarchy is static and irrelevant to text samples, it makes these methods hard to exploit hierarchical information. Contrary to global hierarchy, local hierarchy as a structured labels hierarchy corresponding to each text sample. It is dynamic and relevant to text samples, which is ignored in previous methods. To exploit global and local hierarchies, we propose Hierarchy-guided BERT with Global and Local hierarchies (HBGL), which utilizes the large-scale parameters and prior language knowledge of BERT to model both global and local hierarchies. Moreover, HBGL avoids the intentional fusion of semantic and hierarchical modules by directly modeling semantic and hierarchical information with BERT. Compared with the state-of-the-art method HGCLR, our method achieves significant improvement on three benchmark datasets. Our code is available at http://github.com/kongds/HBGL.

1 Introduction

Hierarchical text classification (HTC) focuses on assigning one or more labels from the label hierarchy to a text sample (Sun and Lim, 2001). As a special case of multi-label text classification, HTC has various applications such as news categorization (Kowsari et al., 2017) and scientific paper classification (Lewis et al., 2004b). The methods in HTC aim to improve prediction accuracy by modeling the large-scale, imbalanced, and structured label hierarchy (Mao et al., 2019a).

To model the label hierarchy, recent methods (Zhou et al., 2020; Chen et al., 2021; Wang et al., 2022) view hierarchy as a directed acyclic graph and model label hierarchy based on graph encoders. However, the input of graph encoders is static, considering that all HTC text samples share the same hierarchical structure, which leads graph encoders to model the same graph redundantly. To solve this problem, Wang et al. (2022) directly discards the graph encoder during prediction, but this method still suffers from the same problem during training. Moreover, since the target labels corresponding to each text sample could be either a single-path or a multi-path in HTC (Zhou et al., 2020), recent methods only consider the graph of the entire label hierarchy and ignore the subgraph corresponding to each text sample. This subgraph can contain structured label co-occurrence information. For instance, a news report about France travel is labeled "European" under the parent label "World" and "France" under a different parent label "Travel Destinations". There is a strong correlation between the labels "France" and "European". But these labels are far apart on the graph, making it difficult for graph encoders to model this relationship.

Under such observation, we divide the label hierarchy into global and local hierarchies to take full advantage of hierarchical information in HTC. We define global hierarchy as the whole hierarchical structure, referred to as hierarchical information in previous methods. Then we define local hierarchy as a structured label hierarchy corresponding to each text sample, which is the subgraph of global hierarchy. Moreover, global hierarchy is static and irrelevant to text samples, while local hierarchy is dynamic and relevant to text samples. Considering the characteristics of two hierarchies, our method models them separately to avoid redundantly modeling static global hierarchy and fully exploit hierarchical information with dynamic local hierarchy.

To model semantic information along with hierarchical information, Zhou et al. (2020) proposes hierarchy-aware multi-label attention. Chen

* Corresponding Author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4030 - 4039
December 7-11, 2022

et al. (2021) reformulates it as a matching problem by encouraging the text representation to be similar to its label representation. Although, these methods can improve the performance of text encoders by injecting label hierarchy with the graph encoder , the improvement on pretrained language model BERT (Devlin et al., 2018) is limited (Wang et al., 2022). Compared to previous text encoders such as CNN or RNN, BERT has large-scale parameters and prior language knowledge. It enables BERT to roughly grasp hierarchical information with multi-label text classification. Therefore, HGCLR (Wang et al., 2022) is proposed to improve the BERT performance on HTC, and the hierarchy is embedded into BERT based on contrastive learning during training. For prediction, HGCLR directly uses BERT as a multi-label classifier. Specifically, the hierarchy in HGCLR is represented by positive samples in contrastive learning, which is implemented by scaling the BERT token embeddings based on a graph encoder. However, representing hierarchy by simply scaling token embeddings is inefficient, which may also lead to a gap between training and prediction.

To efficiently exploit BERT in HTC, we leverage the prior knowledge of BERT by transforming both global and local hierarchy modeling as mask prediction tasks. Moreover, we discard the auxiliary graph encoder and utilize BERT to model hierarchical information to avoid the intentional fusion of BERT and graph encoder. For global hierarchy, we propose a label mask prediction task to recover masked labels based on the label relationship in global hierarchy. Since global hierarchy is irrelevant to text samples, we only fine-tune label embeddings and keep BERT frozen. For local hierarchy, we combine text samples and labels as the input of BERT to directly fuse semantic and hierarchical information according to the attention mechanism in BERT.

In summary, the contributions of this paper are following:

- We propose HBGL to take full advantage of BERT in HTC. HBGL does not require auxiliary modules like graph encoders to model hierarchical information, which avoids the intentional fusion of semantic and hierarchical modules.

- We propose corresponding methods to model the global and local hierarchies based on their characteristics in order to further exploit the information of the hierarchy.

- Experiments show that the proposed model achieves significant improvements on three datasets. Our code will be public to ensure reproducibility.

2 Related Work

Hierarchical text classification (HTC) is a special multi-label text classification problem that requires constructing one or more paths from the taxonomic hierarchy in a top-down manner (Sun and Lim, 2001). Compared to multi-label text classification, HTC focuses on leveraging hierarchical information to achieve better results. There are two groups of existing HTC methods based on treating the label hierarchy: local and global approaches.

The local approaches leverage hierarchical information by constructing one or more classifiers at each level or each node in hierarchy. Generally speaking, a text sample will be classified top-down according to its hierarchy. Shimura et al. (2018) applies a CNN with a fine-tuning technique to utilize the data in the upper levels. Banerjee et al. (2019) initials parameters of the child classifier by the fine-tuned parent classifiers.

The global approaches leverage hierarchical information by directly treating HTC as multi-label text classification with hierarchy information as input. Many methods like recursive regularization (Gopal and Yang, 2013), reinforcement learning (Mao et al., 2019b), capsule network (Peng et al., 2021), and meta-learning (Wu et al., 2019) has been proposed to capture hierarchical information. To better represent hierarchical information, Zhou et al. (2020) formulates the hierarchy as a directed graph and introduces hierarchy-aware structure encoders. Chen et al. (2021) formulates the text-label semantics relationship as a semantic matching problem.

With the development of Pretrained Language Model (PLM), PLM outperforms previous methods even without using hierarchical information. Compared to text encoders like RNN or CNN, PLM is strong enough to learn hierarchical information without hierarchy-aware structure encoders. Under this observation, Wang et al. (2022) proposes HGCLR to embed the hierarchical information into the PLM directly. However, HGCLR still requires the hierarchy-aware structure encoder like

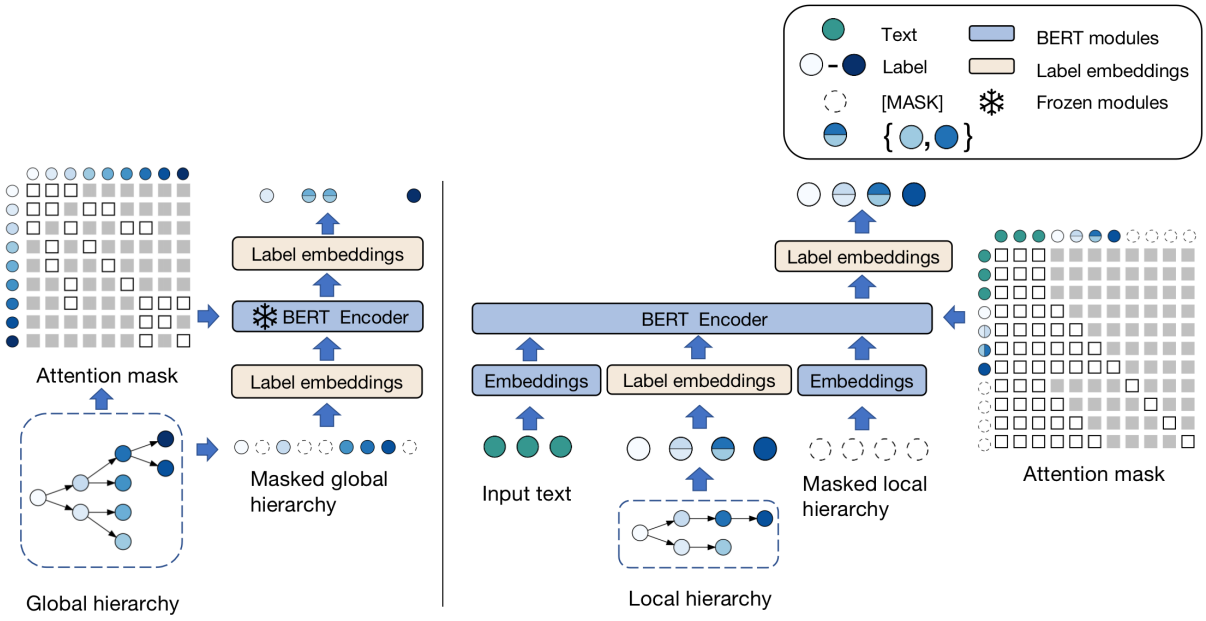

Figure 1: The overall framework of our model under the label hierarchy with four maximum levels. The left part is the global hierarchy-aware label embeddings module. The right part is the local hierarchy-aware text encoder module. We use different colors to identify labels, and the darker color indicates the lower level. Gray squares in attention masks indicate that tokens are prevented from attending, while white squares indicate that attention between tokens is allowed. The label embeddings share the same weight in both modules, which is initialized by the global hierarchy-aware label embeddings module. Note the special tokens like `[CLS]` or `[SEP]`, position and segment tokens of BERT are ignored for simplicity, which we will discuss in methodology. (Best view in color.)

Graphormer (Ying et al., 2021) to incorporate hierarchical information during training.

3 Problem Definition

Given a training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ where $\mathbf{x}_i$ is raw text, and $\mathbf{y}_i \in \{0,1\}^L$ is the label of $\mathbf{x}_i$ represented by L dimensional multi-hot vector. The goal of Hierarchical Text Classification (HTC) is to predict a subset labels for $\mathbf{x}_i$ with the help of hierarchical information, which can be organized as a Directed Acyclic Graph (DAG) $G = (V, \overrightarrow{E}, \overleftarrow{E})$, where $V = \{v_1, \ldots, v_L\}$ is the set of label nodes. $\overrightarrow{E} = \{(v_i, v_j) | v_j \in child(v_i)\}$ is the top-down hierarchy path and $\overleftarrow{E} = \{(v_j, v_i) | v_j \in child(v_i)\}$ is the bottom-up hierarchy path. Although labels in $\mathbf{y}_i$ follow the labels hierarchy, HTC could be either a single-path or a multi-path problem (Zhou et al., 2020).

4 Methodology

In this section, we provide the technical details of the proposed HBGL. Figure 1 shows the overall framework of the model. The left part corresponds to global hierarchy-aware label embeddings, while the right part corresponds to local hierarchy-aware text encoder. We first inject global hierarchy into label embeddings in global hierarchy-aware label embeddings. Then we leverage these label embeddings with our local hierarchy-aware text encoder.

4.1 Global Hierarchy-aware Label Embeddings

Global hierarchy-aware label embeddings aims to initialize the label embeddings based on the label semantics and hierarchy in HTC. It allows BERT to directly exploit the global hierarchy without auxiliary label encoders. Contrary to previous methods (Zhou et al., 2020; Chen et al., 2021; Wang et al., 2022), we implement BERT as a graph encoder to initialize the label embeddings via gradients descent and adapt the global hierarchy to label embeddings by formulating it as mask prediction. Global hierarchy-aware label embeddings leverages the large-scale pretraining knowledge of BERT to generate better hierarchy-aware and semantic-aware label embeddings.

Following Wang et al. (2022), we first initialize label embeddings $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_L] \in \mathbb{R}^{L \times d}$ with the averaging BERT token embeddings of label name, where d is the hidden size of BERT and $\hat{\mathbf{y}}_i$ corresponds to label node v_i in G. Since $\hat{\mathbf{Y}}$

is initialized with the BERT token embedding, it takes advantage of the prior knowledge of BERT to merge label semantic and hierarchical information. Specifically, the input embeddings $\mathbf{e} \in \mathbb{R}^{L\times d}$ of BERT encoder is defined as:

$$\mathbf{e}_i = \hat{\mathbf{y}}_i + \mathbf{t}_1 + \mathbf{p}_{\text{HLevel}(v_i)} \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^{512\times d}$ and $\mathbf{t} \in \mathbb{R}^{2\times d}$ are the position embeddings and segment embeddings in BERT (Devlin et al., 2018). To exploit the position embeddings $\mathbf{p}$, we use the hierarchy level $\text{HLevel}(v_i)$ as the position id for label v_i. Low position ids represent coarse-grained labels, while high position ids represent fine-grained labels. To exploit the segment embeddings $\mathbf{t}$, we use segment id 1 to represent labels, which makes BERT easy to distinguish labels and text in HTC.

To feed BERT with the label graph, we add attention mask $\mathbf{A} \in \{0,1\}^{L\times L}$ in each self-attention layers. Formally, $\mathbf{A}$ is defined as:

$$\mathbf{A}_{ij} = \begin{cases} 0, & \text{if}(v_i, v_j) \in \overrightarrow{E} \cup \overleftarrow{E} \text{ or } i = j \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

where $\overrightarrow{E}$ is the top-down hierarchy path and $\overleftarrow{E}$ is the bottom-up hierarchy path of DAG G. We allow one label can attend its parent and child labels. For example, we show the attention mask for the four-level hierarchy in Figure 1.

Based on input embeddings $\mathbf{E}$ and attention mask $\mathbf{A}$, we can use mask LM task (Devlin et al., 2018) to inject hierarchy into $\hat{\mathbf{Y}}$. However, if we directly follow mask LM task in BERT, it will cause BERT unable to distinguish between masked leaf labels under the same parent label. For example, two leaf labels "Baseball" and "Football" under the same parent label "Sport". The model will output the same result if both leaf labels are masked. Since both labels have the same position and segment embeddings, and only attend to "Sport" label in $\mathbf{A}$. To solve this problem, we treat the masked label prediction task as the multi-label classification, which requires a masked leaf label to predict itself and other masked sibling leaf labels according to G.

Formally, we first random mask several labels by replacing $\hat{\mathbf{y}}_i$ with mask token embedding in Eq. 1 to get masked input embeddings $\mathbf{e}'$. Second, we calculate the hidden state representation $\mathbf{h} \in \mathbb{R}^{L\times d}$ and scores of each label $\mathbf{s} \in \mathbb{R}^{L\times L}$ as following:

$$\begin{aligned} \mathbf{h} &= \text{BERTEncoder}(\mathbf{e}', \mathbf{A}) \\ \mathbf{s} &= \text{sigmoid}(\mathbf{h}\hat{\mathbf{Y}}^T) \end{aligned} \tag{3}$$

Where BERTEncoder is the encoder part of BERT and $\mathbf{A}$ is applied to each layer of BERTEncoder. Finally, The problem of injecting hierarchy into $\hat{\mathbf{Y}}$ can be reformulated as solving the following optimization problem:

$$\min_{\hat{\mathbf{Y}}} \mathcal{L}_{\text{global}} = - \sum_{i}^{i\in V_m} \sum_{j=1}^{L} [\bar{\mathbf{y}}_{ij} \log(\mathbf{s}_{ij}) + (1-\bar{\mathbf{y}}_{ij}) \log(1-\mathbf{s}_{ij})] \tag{4}$$

Where V_m is the masked labels set and $\bar{\mathbf{y}}_{ij}$ is the target for $\mathbf{s}_{ij}$. We set $\bar{\mathbf{y}}_{ij} = 1$, when $i = j$ or the label i and j are masked sibling leaf nodes in G. To avoid model overfitting on static graph G, we keep all parameters of BERT frozen and only fine-tune label embeddings $\hat{\mathbf{Y}}$ in Eq. 4. Moreover, we gradually increase the label mask ratio during training.

The whole procedure of global hierarchy-aware label embeddings is shown in Algorithm 1.

Algorithm 1 Global Hierarchy-aware Label Embeddings

input: Label hierarchy G and label names
output: Label embeddings $\hat{\mathbf{Y}}$.
initialize: $\hat{\mathbf{Y}}$ using averaging BERT token embeddings of each label name.

Set mask ratio r_m;
Set mask ratio upper bound r_M;
Set learning rate lr, batch size bsz and training steps T_{train};
Get attention mask $\mathbf{A}$ according to Eq. 2;
for $t = 1, ..., T_{train}$ **do**
  Get input embeddings e according to Eq. 1;
  for $b = 1, ..., bsz$ **do**
    Mask e with mask ratio r_m^t to get $\mathbf{e}'$;
    Get $\mathbf{h}$ and $\mathbf{s}$ according to Eq. 3;
    Get $\bar{\mathbf{y}}$ based on $\mathbf{e}'$ and G, $\bar{\mathbf{y}}_{ij} = 1$, when $j = i$ or the label i and j are masked sibling leaf nodes;
    Compute loss $\mathcal{L}$ in Eq. 4;
    Backward and compute the gradients $\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{Y}}^{tb}}$;
    Accumulate gradients $\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{Y}}^{tb}}$ to $\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{Y}}^{t}}$
  end for
  $r_m^{t+1} = r_m^t + \frac{r_M - r_m}{T_{train}}$
  $\hat{\mathbf{Y}}^{t+1} = \text{UpdateParameter}(\hat{\mathbf{Y}}^t, \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{Y}}^t}, lr)$;
end for

4.2 Local Hierarchy-aware Text Encoder

Local hierarchy is the structured label hierarchy corresponding to each text sample, which is ignored in previous methods (Zhou et al., 2020; Chen et al., 2021; Wang et al., 2022). In contrast to global hierarchy, local hierarchy is dynamic and related to semantic information. To leverage local hierarchy in HTC, we need to examine several issues before introducing our methods. First, although local hierarchy contains the hierarchical information related to target labels, this leads to label leakage during training. Second, we should pay attention to the gap between training and prediction, since local hierarchy is only available during training. To this end, we propose local hierarchy-aware text encoder to exploit local hierarchy while avoiding the above issues.

4.2.1 Local Hierarchy Representation

We first discuss the representation of local hierarchy in BERT before introducing local hierarchy-aware text encoder. Following the method in global hierarchy-aware label embeddings, we can represent local hierarchy as the subgraph of global hierarchy according to attention mechanism of BERT. However, it is hard for BERT to combine the input of label graph and text sample, while avoiding label leakage and the gap between training and prediction. Therefore, we propose another method to efficiently represent local hierarchy, which allows BERT to combine local hierarchies with text samples easily. Since the local hierarchy is single-path or multi-path in the global hierarchy, in the single-path case we can simply treat it as the sequence. If we can also transform the multi-path case into the sequence, we can represent local hierarchy as the sequence, making it easy to model with BERT. Under this observation, we use the following method to transform the multi-path local hierarchy into the sequence:

$$\mathbf{u}_h = \sum_{j}^{j \in y^h} \hat{\mathbf{y}}_j \qquad (5)$$
$$\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_D]$$

Where $\mathbf{u}_h \in \mathbb{R}^d$ is the hth level of hierarchy, y^h is the target labels in hth level, $\hat{\mathbf{y}}_j$ is the global hierarchy-aware label embeddings, D is the maximum level of hierarchy and $\mathbf{u}$ is local hierarchy sequence.

For example, consider a multi-path local hierarchy with four labels: $1a$, $1b$, $2a$ and $2b$, where $2a$ and $2b$ are the child labels of $1a$ and $1b$, respectively. According to Eq. 5, we can get $\mathbf{u}_1 = \hat{\mathbf{y}}_{1a} + \hat{\mathbf{y}}_{1b}$ and $\mathbf{u}_2 = \hat{\mathbf{y}}_{2a} + \hat{\mathbf{y}}_{2b}$. The attention score α_{21} in BERT can be calculated as:

$$\begin{aligned}\alpha_{21} =& \mathbf{u}_2 W^Q (\mathbf{u}_1 W^K)^T \\ =& \hat{\mathbf{y}}_{2a} W^Q (\hat{\mathbf{y}}_{1a} W^K)^T + \hat{\mathbf{y}}_{2b} W^Q (\hat{\mathbf{y}}_{1b} W^K)^T \\ &+ \hat{\mathbf{y}}_{2a} W^Q (\hat{\mathbf{y}}_{1b} W^K)^T + \hat{\mathbf{y}}_{2b} W^Q (\hat{\mathbf{y}}_{1a} W^K)^T\end{aligned} \qquad (6)$$

Where $W^Q, W^K \in \mathbb{R}^{d \times d_z}$ are parameter matrices in BERT. As shown in Eq. 6, α_{21} contains $\hat{\mathbf{y}}_{2a} W^Q (\hat{\mathbf{y}}_{1a} W^K)^T$ and $\hat{\mathbf{y}}_{2b} W^Q (\hat{\mathbf{y}}_{1b} W^K)^T$ to represent the local hierarchy graph, while it also contains $\hat{\mathbf{y}}_{2a} W^Q (\hat{\mathbf{y}}_{1b} W^K)^T$ and $\hat{\mathbf{y}}_{2b} W^Q (\hat{\mathbf{y}}_{1a} W^K)^T$. Since we have injected global hierarchy into the label embeddings: $\hat{\mathbf{y}}_{1a}$, $\hat{\mathbf{y}}_{1b}$, $\hat{\mathbf{y}}_{2a}$ and $\hat{\mathbf{y}}_{2b}$ according to Algorithm 1, which leverages the first part in α_{21} to predict masked labels, it allows α_{21} to be able to hold hierarchical information in local hierarchy. In addition, the second part of α_{21} is also relevant for modeling the local hierarchy, as the labels in it correspond to the same text sample.

4.2.2 Fusing Local Hierarchy into Text Encoder

To further exploit local hierarchy, while avoiding label leakage and the gap between training and prediction, it is hard to implement BERT directly as multi-label classifier. Inspired by s2s-ft (Bao et al., 2021), which adopts PLM like BERT for sequence-to-sequence learning, we propose a novel method by adopting BERT to generate the local hierarchy sequence. Note that since elements in the local hierarchy sequence may contain multiple labels, we cannot use sequence-to-sequence methods directly.

In order to fuse local hierarchy and text in sequence-to-sequence fashion, our model aims to generate the local hierarchy sequence $\mathbf{u}$:

$$p(\mathbf{u} \mid \mathbf{x}) = \prod_{h=1}^{D} p\left(\mathbf{u}_h \mid \mathbf{u}_{<h}, \mathbf{x}\right) \qquad (7)$$

where $\mathbf{u}_{<h} = \mathbf{u}_1, \ldots, \mathbf{u}_{h-1}$ and $\mathbf{x}$ is the input text corresponding to $\mathbf{u}$. There are several advantages to model HTC as Eq. 7: First, the structure of local hierarchy can be included. Since $\mathbf{u}_h$ represents the labels corresponding to the hth level of hierarchy, it only depends on the labels above hth level, which

can been represented by $p\left(\mathbf{u}_h \mid \mathbf{u}_{<h}, x\right)$. Second, we can leverage the teacher forcing to fuse local hierarchy and text while avoiding label leakage during training.

Specifically, the input of BERT is composed of three parts: input text, local hierarchy and masked local hierarchy during training, as shown in Figure 1. The end-of-sequence token `[SEP]` is used to divide these three parts. Based on s2s-ft, we implement similar attention mask provided in Figure 1. The attention mask prevents the input text from attending to the local hierarchy and masked local hierarchy, which guarantees that labels do not influence the input text tokens. The attention mask also ensures the top-down manner in the local hierarchy, which allows the label to attend to the upper level labels in the hierarchy. For the attention mask between local hierarchy and masked local hierarchy, it allows the masked labels to be predicted based on upper level target labels, following the teacher forcing manner. We use 0 and 1 to distinguish text and label for segment ids in BERT. For position ids in BERT, we set the same position ids in the local hierarchy and mask local hierarchy by accumulating them based on text location ids. By feeding BERT with the above inputs, we use a binary cross-entropy loss function to predict labels in each level separately. The optimization problem is as follows:

$$\min_{\Theta,\hat{\mathbf{Y}}} \mathcal{L}_{\text{local}} = -\sum_{i=1}^{N}\sum_{h=1}^{D}\sum_{j}^{j\in V_h}\left[\mathbf{y}_{ij}\log\left(\mathbf{s}_{ihj}^{t}\right)+\left(1-\mathbf{y}_{ij}\right)\log\left(1-\mathbf{s}_{ihj}^{t}\right)\right] \tag{8}$$

where Θ are parameters of BERT, $V_h = \{j \mid \text{HLevel}(v_j) = h\}$ is the labels in hth level, $\mathbf{y}_{ij}$ is the jth label corresponding to text $\mathbf{x}_i$ and $\mathbf{s}_{ihj}^{t}$ is the score of jth label, which is calculated as following:

$$\mathbf{s}_{ih}^{t} = \text{sigmoid}(\mathbf{h}_{ih}^{t}\hat{\mathbf{Y}}^{T}) \tag{9}$$

where $\mathbf{h}_{ih}^{t} \in \mathbb{R}^d$ is the hidden state representation of hth masked local hierarchy corresponding to $\mathbf{x}_i$ and $\mathbf{s}_{ihj}^{t}$ is the jth element of $\mathbf{s}_{ih}^{t} \in \mathbb{R}^L$.

For prediction, we utilize the local hierarchy during the training stage to make BERT separately predict labels at each level in an autoregressive manner. The scores $\mathbf{s}_i^p$ for ith level labels are computed as following:

$$\begin{aligned}\mathbf{h}_h^p &= \text{BERTEncoder}([\mathbf{e}_{\text{text}}; \mathbf{u}_{<h}^p; \mathbf{e}_{\text{mask}}], \mathbf{A}^{p_h})\\ \mathbf{s}_h^p &= \text{sigmoid}(\mathbf{h}_h^p\hat{\mathbf{Y}}^T)\end{aligned} \tag{10}$$

where $\mathbf{h}_h^p \in \mathbb{R}^d$ is the hidden state representation of hth level, $\mathbf{e}_{\text{text}}$ and $\mathbf{e}_{\text{mask}}$ are text embeddings and mask token embedding, $\mathbf{A}^{p_h}$ is the attention mask for hth level, which is a submatrix of the attention mask between text and local hierarchy in training, and $\mathbf{u}_{<h}^p = \mathbf{u}_1^p \cdots \mathbf{u}_{h-1}^p$ is:

$$\begin{aligned}\mathbf{u}_h^p &= \begin{cases}\mathbf{e}_{\text{sep}}, & \text{if } \sum_{j=0}^{L}\mathbb{1}(\mathbf{s}_{hj}^p) = 0\\ \sum_{j=0}^{L}\mathbb{1}(\mathbf{s}_{hj}^p)\hat{\mathbf{y}}_j^T, & \text{otherwise}\end{cases}\\ s.t\mathbb{1}\left(\mathbf{s}_{hj}^p\right) &= \begin{cases}1, & \text{if } \mathbf{s}_{hj}^p > 0.5\\ 0, & \text{otherwise}\end{cases}\end{aligned} \tag{11}$$

where $\mathbb{1}\left(\mathbf{s}_{hj}^p\right)$ represents the predicted label corresponding to $\mathbf{s}_{hj}^p$. Specifically, we sum hth level predicted label embeddings to generate $\mathbf{u}_h^p$ and replace $\mathbf{u}_h^p$ with $\mathbf{e}_{\text{sep}}$ `[SEP]` token embedding when these is no predicted label in hth level.

Finally, the predicted labels set y^p is:

$$y^p = \{v_j | \mathbf{s}_{hj}^p > 0.5 \text{ and } h = \text{HLevel}(v_j)\} \tag{12}$$

5 Experiments

Datasets and Evaluation Metrics We select three widely-used HTC benchmark datasets in our experiments. They are: Web-of-Science (WOS) (Kowsari et al., 2017), NYTimes (NYT) (Shimura et al., 2018), and RCV1-V2 (Lewis et al., 2004a). The detailed information of each dataset is shown in Table 1. We follow the data processing of previous works (Zhou et al., 2020; Wang et al., 2022) and use the same evaluation metrics to measure the experimental results: Macro-F1 and Micro-F1.

Dataset	L	D	Avg(L_i)	Train	Dev	Test
WOS	141	2	2.0	30,070	7,518	9,397
NYT	166	8	7.6	23,345	5,834	7,292
RCV1-V2	103	4	3.24	20,833	2,316	781,265

Table 1: Data Statistics. L is the number of classes. D is the maximum level of hierarchy. Avg($|L_i|$) is the average number of classes per sample.

Model	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Hierarchy-Aware Models						
TextRCNN (Zhou et al., 2020)	83.55	76.99	70.83	56.18	81.57	59.25
HiAGM (Zhou et al., 2020)	85.82	80.28	74.97	60.83	83.96	63.35
HTCInfoMax (Deng et al., 2021)	85.58	80.05	-	-	83.51	62.71
HiMatch (Chen et al., 2021)	86.20	80.53	-	-	84.73	64.11
Pretrained Language Models						
BERT†	85.63	79.07	78.24	65.62	85.65	67.02
BERT+HiAGM†	86.04	80.19	78.64	66.76	85.58	67.93
BERT+HTCInfoMax†	86.30	79.97	78.75	67.31	85.53	67.09
BERT+HiMatch (Chen et al., 2021)	86.70	81.06	-	-	86.33	68.66
HGCLR (Wang et al., 2022)	87.11	81.20	78.86	67.96	86.49	68.31
HBGL	**87.36**	**82.00**	**80.47**	**70.19**	**87.23**	**71.07**

Table 2: Experimental results of our proposed model on several datasets. †: results from (Wang et al., 2022).

5.1 implement Details

Following HGCLR (Wang et al., 2022), we use `bert-base-uncased` as both text and graph encoders. The graph structure input of BERT is implemented based on the attention mask of huggingface transformers (Wolf et al., 2020). We introduce the implementation details of the global hierarchy-aware label embeddings and the local hierarchy-aware text encoder, respectively.

For global hierarchy-aware label embeddings, we first initialize the label embeddings by averaging their label name token embeddings in BERT. To embed global hierarchy into label embeddings, we train initialized label embeddings with frozen `bert-base-uncased` according to Algorithm 1. The initial mask ratio and mask ratio upper bound are 0.15 and 0.45, respectively. Considering the different maximum levels of hierarchy and the number of labels in each dataset, we grid search learning rates of label embeddings among {1e-3, 1e-4} and the training steps among {300, 500, 1000}.

For local hierarchy-aware text encoder, we follow the settings of HGCLR. The batch size is set to 12. The optimizer is Adam with a learning rate of 3e-5. We use global hierarchy-aware label embeddings to initialize the label embeddings. And the input label embeddings share weights with the label embeddings in the classification head. The input label length of each dataset is set to the depth in Table 1. Compared to the 512 maximum input tokens in HGCLR, we use smaller input tokens to achieve similar prediction time performance. According to the maximum hierarchy level, the maximum input tokens in WOS, NYT, and RCV1-V2 are 509, 472, and 492, respectively. Moreover, we cache the previous attention query and key values to make the prediction more efficient.

5.2 Baselines

We compared the state-of-the-art and most enlightening methods including HiAGM (Zhou et al., 2020), HTCInfoMax (Deng et al., 2021), HiMatch (Chen et al., 2021), and HGCLR (Wang et al., 2022). HiAGM, HTCInfoMax, and HiMatch use different fusion strategies to mix text-hierarchy representation. Specifically, HiAGM proposes hierarchy-aware multi-label attention to get the tex-hierarchy representation. HTCInfoMax introduces information maximization to model the interaction between text and hierarchy. HiMatch reformulates it as a matching problem by encouraging the text representation be similar to its hierarchical label representation. Contrary to the above methods, HGCLR achieves state-of-the-art results by directly incorporating hierarchy into BERT based on contrastive learning.

5.3 Experimental Results

Table 2 shows Micro-F1 and Macro-F1 on three datasets. Our method significantly outperforms all methods by further exploiting the hierarchical information of HTC and the prior knowledge of BERT.

Compared to BERT, we show proposed HBGL

can significantly leverage the hierarchical information by achieving 2.99%, 4.59% and 4.05% improvement of Macro-F1 on WOS, NYT, and RCV1-V2. By the way, our method shows better performance on the dataset with a complex hierarchy. Our method can achieve 4.59% improvement of Macro-F1 on NYT with the largest label depth and number of labels in the three datasets.

HBGL also shows impressive performance compared to BERT based HTC methods. Current state-of-the-art methods like HGCLR, which relies on contrastive learning to embed the hierarchy into BERT, has negligible improvement over previous methods such as HiMatch. Furthermore, although different methods of incorporating semantic and hierarchical information are used, these methods have similar performances on three datasets, which shows a common limitation in previous methods: merging BERT and the graph encoder regardless of local hierarchy and prior knowledge of BERT. By overcoming this limitation, our method observes 2.23% and 2.76% boost on Macro-F1 on NYT and RCV1-V2 compared to HGCLR. For WOS, it is the simplest dataset among the above datasets with two-level label hierarchy, and the labels for each document are single-path in the hierarchy, the impact of leveraging hierarchy is smaller than the other two datasets. However, our methods still achieve reasonable improvement compared to HGCLR.

5.4 Effect of Global Hierarchy-aware Label Embeddings

To examine the effectiveness of global hierarchy-aware label embeddings, we compare four label embeddings methods: global hierarchy-aware label embeddings (global hierarchy BERT), global hierarchy-aware label embeddings, where BERT is replaced with GAT (global hierarchy GAT), label embeddings initialized by averaging BERT token embeddings in each label name (label name) and randomly initialize label embeddings (random). For global hierarchy GAT and BERT, we use label names as the initialized label embeddings. For global hierarchy GAT, we get the best results by grid searching learning rate, training step and whether to freeze GAT.

As shown in Table 3, our method outperforms the other three label embeddings methods. Although all methods can leverage hierarchical information by the local hierarchy-aware text encoder, the remaining methods still achieve poorer performance than global hierarchy BERT, which shows the importance of global hierarchy. For global hierarchy GAT, GAT cannot leverage global hierarchy by predicting masked labels like BERT.

Label Embeddings	NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Random	79.18	67.92	86.98	70.15
Label name	80.26	69.64	87.20	70.24
Global hierarchy GAT	80.15	69.59	86.69	70.23
Global hierarchy BERT	**80.47**	**70.19**	**87.23**	**71.07**

Table 3: Impact of different label embeddings on NYT and RCV1-V2.

5.5 Effect of Local Hierarchy-aware Text Encoder

We also analyze the importance of local hierarchy-aware text encoder by comparing it with two methods: multi-label and seq2seq. For multi-label, we fine-tune BERT as the multi-label classifier. For seq2seq, we fine-tune BERT as the seq2seq model following s2s-ft, where target labels are sorted according to their levels in the global hierarchy. Local hierarchy-aware text encoder achieves the best performance in Table 4.

Fine tuning	NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Mulit-label	78.16	67.05	85.96	68.03
Seq2seq	79.22	67.82	86.22	67.97
Local hierarchy	**80.47**	**70.19**	**87.23**	**71.07**

Table 4: Impact of different fine tuning methods on NYT and RCV1-V2.

6 Conclusion

In this paper, we proposed a BERT-based HTC framework HBGL. HBGL avoids the intentional fusion of semantic and hierarchical modules by utilizing BERT to model both semantic and hierarchical information. Moreover, HBGL takes full advantage of hierarchical information by modeling global and local hierarchies, respectively. Considering that global hierarchy is static and irrelevant to text samples, we propose global hierarchy-aware label embeddings to inject global hierarchy into label embeddings directly. Considering that local hierarchy is dynamic and relevant to text samples, we propose local hierarchy-aware text encoder to deeply combine semantic and hierarchical information according to the attention mechanism in BERT.

Compared to existing methods, HBGL achieves significant improvements on all three datasets, while only parameters corresponding to label embeddings are required except BERT.

7 Limitation

While HBGL exploits global and local hierarchies and achieves improvements on three HTC datasets, one limitation is that HBGL requires additional iterations to predict labels. HBGL needs to predict upper level labels before predicting current level labels. To alleviate this limitation, we cached the BERT attention query and key values from previous iterations and used a smaller source length than HGCLR, which allowed HBGL to achieve similar inference speeds compared to HGCLR. Specifically, HGCLR achieves 1.02× to 1.10× inference speedups over HBGL on three datasets.

8 Acknowledgments

The research work is supported by the National Key Research and Development Program of China under Grant No. 2019YFA0707204, the National Natural Science Foundation of China under Grant Nos. 62276015, 62176014, the Fundamental Research Funds for the Central Universities.

References

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300, Florence, Italy. Association for Computational Linguistics.

Hangbo Bao, Li Dong, Wenhui Wang, Nan Yang, and Furu Wei. 2021. s2s-ft: Fine-tuning pretrained transformer encoders for sequence-to-sequence learning. *arXiv preprint arXiv:2110.13640*.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware label semantics matching network for hierarchical text classification. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 4370–4379.

Zhongfen Deng, Hao Peng, Dongxiao He, Jianxin Li, and Philip Yu. 2021. HTCInfoMax: A global model for hierarchical text classification via information maximization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3259–3265, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Siddharth Gopal and Yiming Yang. 2013. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 257–265, New York, NY, USA. Association for Computing Machinery.

Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004a. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004b. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019a. Hierarchical text classification with reinforced label assignment. *arXiv preprint arXiv:1908.10419*.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019b. Hierarchical text classification with reinforced label assignment. 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)*, pages 445–455, Hong Kong, China. Association for Computational Linguistics.

Hao Peng, Jianxin Li, Senzhang Wang, Lihong Wang, Qiran Gong, Renyu Yang, Bo Li, Philip S. Yu, and Lifang He. 2021. Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2505–2519.

Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. HFT-CNN: Learning hierarchical category structure for multi-label short text categorization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 811–816, Brussels, Belgium. Association for Computational Linguistics.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 521–528. IEEE.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. Incorporating Hierarchy into Text Encoder: a Contrastive Learning Approach for Hierarchical Text Classification. *arXiv preprint arXiv:2203.03825*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jiawei Wu, Wenhan Xiong, and William Yang Wang. 2019. Learning to learn and predict: A meta-learning approach for multi-label 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)*, pages 4354–4364, Hong Kong, China. Association for Computational Linguistics.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117, Online. Association for Computational Linguistics.