HPT: Hierarchy-aware Prompt Tuning for Hierarchical Text Classification

Zihan Wang[1†] **Peiyi Wang**[1†] **Tianyu Liu**[2] **Binghuai Lin**[2]
Yunbo Cao[2] **Zhifang Sui**[1] **Houfeng Wang**[1*]
[1] MOE Key Laboratory of Computational Linguistics, Peking University, China
[2] Tencent Cloud Xiaowei
{wangzh9969, wangpeiyi9979}@gmail.com; {szf, wanghf}@pku.edu.cn
{rogertyliu, binghuailin, yunbocao}@tencent.com;

Abstract

Hierarchical text classification (HTC) is a challenging subtask of multi-label classification due to its complex label hierarchy. Recently, the pretrained language models (PLM) have been widely adopted in HTC through a fine-tuning paradigm. However, in this paradigm, there exists a huge gap between the classification tasks with sophisticated label hierarchy and the masked language model (MLM) pretraining tasks of PLMs and thus the potential of PLMs cannot be fully tapped. To bridge the gap, in this paper, we propose **HPT**, a **H**ierarchy-aware **P**rompt **T**uning method to handle HTC from a multi-label MLM perspective. Specifically, we construct a dynamic virtual template and label words that take the form of soft prompts to fuse the label hierarchy knowledge and introduce a zero-bounded multi-label cross-entropy loss to harmonize the objectives of HTC and MLM. Extensive experiments show HPT achieves state-of-the-art performances on 3 popular HTC datasets and is adept at handling the imbalance and low resource situations. Our code is available at https://github.com/wzh9969/HPT.

1 Introduction

Hierarchical text classification (HTC) aims to categorize a text into a set of labels with a structured class hierarchy (commonly modeled as a tree) (Silla and Freitas, 2011). HTC is a multi-label text classification problem, where the classification result corresponds to one or more paths of the hierarchy (Zhou et al., 2020). The major challenge of HTC is to model the large-scale, imbalanced, and structured label hierarchy (Mao et al., 2019).

As shown in Figure 1(a), existing state-of-the-art HTC models (Zhou et al., 2020; Deng et al., 2021; Chen et al., 2021; Zhao et al., 2021) separately extract text and label hierarchy features by utilizing text and graph encoders, and then fuse the two sources of features into a final representation for text classification. Specifically, Chen et al. (2021) takes advantage of powerful pretrained language models (PLMs) in HTC through a fine-tuning paradigm, where they use PLMs as the text encoder. In this paradigm, the PLMs are trained to infer with complex label hierarchy.

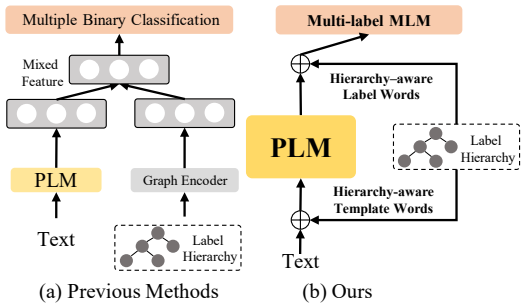


Figure 1: Comparison of previous methods and our HPT. (a) Previous models formulate HTC as a multiple binary classification problem and utilize the PLM in a fine-tuning paradigm. (b) HPT follows a prompt tuning paradigm that transforms HTC into a hierarchy-aware multi-label MLM problem.

Despite the success of the fine-tuning paradigm, some recent studies suggest that it may suffer from distinct training strategies in the pretraining and fine-tuning stages, which restrains the fine-tuned models to take full advantage of knowledge in PLMs (Chen et al., 2022). Therefore, a new paradigm known as *prompt tuning* is proposed to bridge the gap between the downstream tasks and the pretraining tasks of PLMs, which can tap the full potential of PLMs. By warping the text (e.g., "**x**") into the model input (e.g., "**x** is `[MASK]`") and taming the PLMs to complete the masked cloze test, prompt tuning has achieved promising performances on the flat text classification where labels have no hierarchy (Shin et al., 2020).

How about the performances of the prompt tuning in HTC? In the pilot study, we test flat prompt

[†]Equal contribution.
[*]Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3740 - 3751
December 7-11, 2022

tuning methods on HTC and surprisingly find that they are even comparable with the state-of-the-art models in HTC. This result suggests that the expressive power of PLMs has been undermined in the prior HTC methods due to the pretraining-finetuning gap. Although the flat prompt tuning methods have somewhat narrowed the gap, there still remain two challenges while combining PLMs with HTC.

1. **hierarchy and flat gap**. Labels of HTC lie on a sophisticated hierarchy while MLM pretraining and flat prompt tuning do not take label hierarchy into consideration.

2. **multi-label and multi-class gap**. HTC is a multi-label classification problem where the output labels are interconnected with a hierarchy while MLM pretraining is formulated as a multi-class classification.

To bridge these two gaps, as shown in Figure 1(b), we propose a hierarchy-aware prompt tuning (HPT) method that solves HTC from a multi-label MLM perspective. In detail, to bridge the hierarchy and flat gap, we incorporate the label hierarchy knowledge into soft prompt with continuous representation. Specifically, we incorporate the depth and width information in the label hierarchy into different virtual template words, which is helpful to alleviate the label imbalance problem as verified by our experiments. To bridge the multi-label and multi-class gap, we transform HTC into a multi-label MLM problem by a zero-bounded multi-label cross-entropy loss which continually seeks to increase the score of the correct label and decrease the score of the incorrect labels.

We summarize our contributions as follows:

- We propose a hierarchy-aware prompt tuning (HPT) method for hierarchical text classification. To the best of our knowledge, this is the first investigation on flat and hierarchical prompt tuning in HTC.

- We summarize two challenging gaps between HTC and masked language modeling (MLM). To bridge these gaps, we transform HTC into a hierarchy-aware multi-label MLM problem.

- Extensive experiments demonstrate that our proposed model achieves new state-of-the-art results on three popular datasets, and is adept at handling label imbalance and low resource situations.

2 Related Work

2.1 Hierarchical Text Classification

Hierarchical text classification (HTC) is a challenging task due to its large-scale, imbalanced, and structured label hierarchy (Mao et al., 2019). Existing work for HTC could be categorized into local and global approaches based on their ways of utilizing the label hierarchy (Zhou et al., 2020): local approaches build classifiers for each node or level while global ones build only one classifier for the entire graph. Although early works on HTC mainly focus on local approaches (Wehrmann et al., 2018; Shimura et al., 2018; Banerjee et al., 2019), global approaches soon become mainstream. The early global approaches neglect the hierarchical structure of labels and view the problem as a flat multi-label classification (Johnson and Zhang, 2015). Later on, some work try to coalesce the label structure by meta-learning (Wu et al., 2019), reinforcement learning (Mao et al., 2019), and attention module (Zhang et al., 2021). Although such methods can capture the hierarchical information, Zhou et al. (2020) demonstrate that encoding the holistic label structure directly by a structure encoder can further improve performance. Following this research, a bunch of models try to study how the hierarchy should interact with the text. Both Chen et al. (2020) and Chen et al. (2021) embed word and label hierarchy jointly in a same space. Deng et al. (2021) constrains label representation with information maximization. Zhao et al. (2021) designs a self-adaption fusion strategy to extract features from text and labels. Wang et al. (2022) adopts contrastive learning to directly inject hierarchical knowledge into the text encoder.

2.2 Prompt tuning

Prompt tuning (Schick and Schütze, 2021) aims to transform the downstream NLP task into the pretraining task of the pretrained language models (PLM), which can bridge their gap and better utilize PLM. The most popular pretraining task of PLM is MLM (Devlin et al., 2019), which masks some words in the input text and requires PLM to recover these masked words. The prompt tuning methods can be broadly divided into 2 categories: (1) *Hard prompt* (Gao et al., 2021; Schick and Schütze, 2021). The hard prompt methods select a template and label words from the vocabulary of PLM, which require carefully manual designing. (2) *Soft prompt* (Hambardzumyan et al., 2021; Qin

and Eisner, 2021). Soft prompt methods first create some continuous vectors as a template and label embeddings and then find the best prompt using the training examples, which eliminates the need for manually-designed prompts.

3 Preliminaries

3.1 Problem Definition

For each hierarchical text classification (HTC) dataset, we have a predefined label hierarchy $\mathcal{H} = (\mathcal{Y}, E)$, where $\mathcal{Y}$ is the label set (also the node set of $\mathcal{H}$) and E is the edge set. In HTC, given an input text $\mathbf{x}$, the models aim to categorise it into a label set $Y \subseteq \mathcal{Y}$. Specifically, we focus on a setting where every node except the root has one and only one father so that the hierarchy can be simplified as a tree-like structure. In this case, labels can be organized into layers where labels in the same layer have the same depth in the tree. The predicted label set Y corresponds to one or more paths in $\mathcal{H}$.

3.2 Vanilla Fine Tuning for HTC

Given an input text $\mathbf{x}$, the vanilla Fine Tuning method first converts it to "`[CLS]` $\mathbf{x}$ `[SEP]`" as the model input, and then utilizes the PLM to encode it. After that, it utilizes $\mathbf{h}_{\texttt{[CLS]}}$, the hidden state of "`[CLS]`", to predict the labels of the input text. Previous methods (Chen et al., 2021; Wang et al., 2022) based on the PLM all follow this fine-tuning paradigm.

3.3 Prompt Tuning for HTC

To bridge the gap between the pretraining task and the downstream tasks, prompt tuning has been proposed. We adopt 2 typical flat text classification prompt methods to HTC.

Hard Prompt For a text "$\mathbf{x}$", hard prompt first applies a template and fills the input into it. For HTC, we choose "`[CLS]` $\mathbf{x}$ `[SEP]` The text is about `[MASK]` `[SEP]`" as template. The PLM is then asked to predict the "`[MASK]`" slot, which outputs a score for every word in the vocabulary. A verbalizer is then selected for each label to represent its meaning: the score of filling that verbalizer into the "`[MASK]`" slot is the prediction score of the corresponding label. We select the headword (the root word on the dependency tree) of the label name as a verbalizer to represent the corresponding label.

Soft Prompt For a text "$\mathbf{x}$", soft prompt append a fixed number of learnable virtual template words to the text (i.e., "`[CLS]` $\mathbf{x}$ `[SEP]` `[V1]` `[V2]` ... `[V8]` `[MASK]` `[SEP]`" in case of 8) as template. During training, the PLM learns to predict the "`[MASK]`" slot as well as tunes virtual template words. For HTC, we create a learnable label embedding as a verbalizer for each hierarchical label.

Since HTC is a multi-label classification problem, following previous works, both the vanilla fine-tuning and 2 typical prompt tuning methods finally conduct multiple binary classifications. The output of PLM is normalized by sigmoid instead of the original softmax to predict each label and the loss function is changed to binary cross-entropy.

Although we can modify these 2 typical prompt tuning methods for HTC, the essence of this challenge has not been considered. As mentioned, the existing prompt methods experience two major gaps when migrating to HTC:

1. **Hierarchy and flat gap.** Both soft prompt and hard prompt do not take labels into account until prediction, and PLM views all candidate words as equal. Previous works suggest that incorporating label dependency instead of modeling them as flat classification is essential for alleviating the label imbalance (Gopal and Yang, 2013).

2. **Multi-label and multi-class gap.** Previous works on HTC view the problem as multiple binary classifications but MLM is designed for multi-class classification. Prompting aims to bridge the gap between pretraining and fine-tuning but the gap still exists if we use the sigmoid normalization and binary cross-entropy loss functions for HTC during fine-tuning.

4 Methodology

In this section, we introduce a hierarchy-aware prompt tuning method to solve HTC from a multi-label MLM perspective.

4.1 Hierarchy-aware Prompt

To bridge the *hierarchy and flat gap*, we create the prompt with the label hierarchy constraint and injection.

4.1.1 Hierarchy Constraint

To incorporate the label hierarchy, we propose a layer-wise prompt. Since the label hierarchy is a

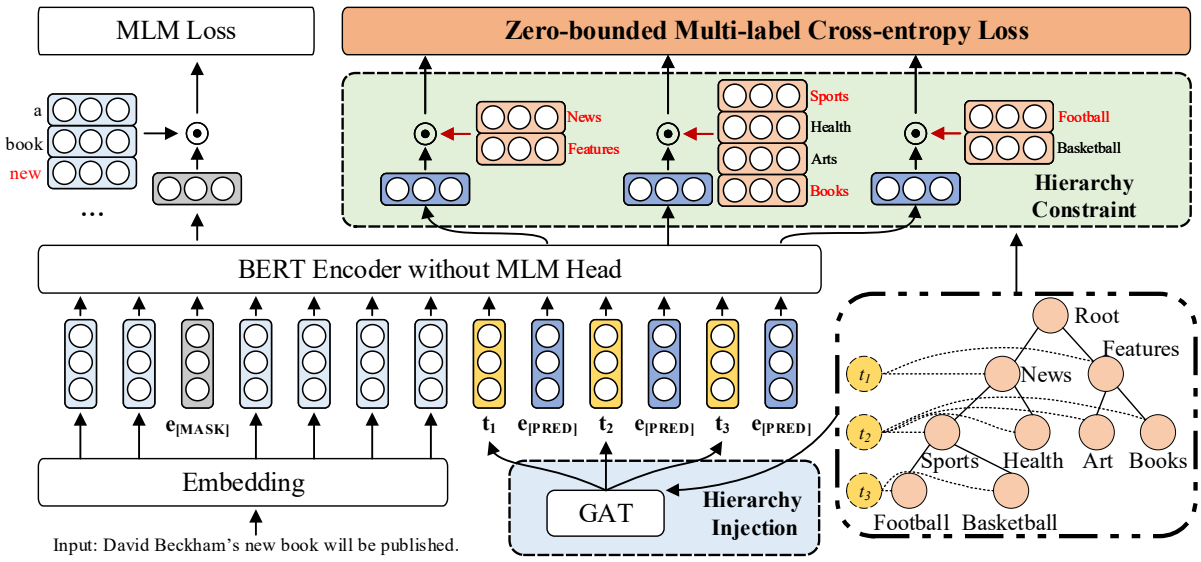


Figure 2: The architecture of HPT during training. HPT transforms HTC into a hierarchy-aware multi-label MLM problem that focuses on bridging *two* gaps between HTC and MLM. (1) To bridge the hierarchy and flat gap, HPT incorporates the label hierarchy knowledge into a dynamic virtual template and label words construction. (2) To bridge the multi-label and multi-class gap. HPT transforms HTC into a multi-label MLM task with a zero-bounded multi-label cross-entropy loss.

tree, we construct templates based on the depth of the hierarchy. Given a predefined label hierarchy $\mathcal{H} = (\mathcal{Y}, E)$ with a depth of L and input text $\mathbf{x}$, the template is "`[CLS]` $\mathbf{x}$ `[SEP] [V1] [PRED] [V2] [PRED] ... [VL] [PRED] [SEP]`". Instead of a fixed number of template words as soft prompts, we have a dynamic template that has template words (from `[V1]` to `[VL]`) the same number as hierarchy layers. We use a special `[PRED]` token for label prediction, indicating a multi-label prediction.

We use BERT (Devlin et al., 2019) as text encoder, which first embeds input tokens to embedding space:

$$\mathbf{T} = [\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{t}_1, \mathbf{e}_P, \ldots, \mathbf{t}_L, \mathbf{e}_P] \quad (1)$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ is word embeddings and $\mathbf{e}_P$ is the embedding of `[PRED]`, which is initialized by the `[MASK]` token of BERT. $\{\mathbf{t_i}\}_{i=1}^{L}$ are layer-wise template embeddings. Similar to soft prompt, template embeddings are randomly initialized and are learned through training. Here we omit special tokens of BERT (`[CLS]` and `[SEP]`) for clarity.

BERT then encodes $\mathbf{T}$ to achieve the hidden states:

$$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N, \mathbf{h}_{t_1}, \mathbf{h}_P^1, \ldots, \mathbf{h}_{t_L}, \mathbf{h}_P^L] \quad (2)$$

where $\mathbf{h}_P^i$ is the hidden state of the i-th $\mathbf{e}_P$, which corresponds to the i-th layer of the label hierarchy.

For verbalizer, we create a learnable virtual label word v_i for each label y_i and initialize its embedding $\mathbf{v}_i$ with the averaging embedding of its corresponding tokens. Instead of predicting all labels in one slot, as shown in the green part of Figure 2, we divide labels into different groups according to their layers and constrain `[PRED]` to only predict labels on one layer. To this end, each template word `[Vi]` is followed by a `[PRED]` token for predictions on the i-th layer. By splitting predictions into different slots, the model may learn better about the dependency between labels across different layers and somewhat solve the label imbalance.

Formally, for $\mathbf{h}_P^m$, we define a verbalizer Verb_m as follows:

$$\text{Verb}_m(y_i) = \begin{cases} v_i, & y_i \in \mathcal{N}_m \\ \varnothing, & \text{Others} \end{cases} \quad (3)$$

where $\mathcal{N}_m$ is the label set of the m-th layer and $\varnothing$ denotes that there is no label word for labels at other layers.

4.1.2 Hierarchy Injection

The hierarchy constraint only introduces the depth of labels but lacks their connectivity. To make full use of the label hierarchy in an MLM manner, we further inject the per-layer label hierarchy knowledge into template embedding.

As shown in the blue part of Figure 2, a K-layer stacked Graph Attention Network (GAT) (Kipf and Welling, 2017) is adopted to model the label hierarchy. Given a node u at the k-th GAT layer, the information interaction and aggregation operation is defined as follows:

$$\mathbf{g}_u^{(k+1)} = \mathrm{ReLU}(\sum_{v \in \mathcal{N}(u) \bigcup \{u\}} \frac{1}{c_u} \mathbf{W}^{(k)} \mathbf{g}_v^{(k)}) \quad (4)$$

where $\mathcal{N}(u)$ denotes the neighbors for node u , c_u is a normalization constant and $\mathbf{W}^{(l)} \in \mathbb{R}^{d_m \times d_m}$ is the trainable parameter.

To achieve per layer knowledge for our layer-wise prompt, we create L virtual nodes $t_1, \ldots, t_L$ (colored in yellow) and connect t_i with all label nodes at the i-th layer in $\mathcal{H}$. In this way, these virtual nodes can aggregate information from a certain hierarchical level through artificial connections. For the first GAT layer, we adopt the virtual label word $\mathbf{v_i}$ for node $y_i \in \mathcal{Y}$ as its node feature and assign template embedding $\mathbf{t_i}$ to virtual node t_i as its node feature.

GAT is then applied to the new graph and it outputs representations $\mathbf{g}_{t_i}^K$ for virtual node t_i, which has gathered knowledge from the i-th layer. We utilize a residual connection to achieve the i-th graph template embedding:

$$\mathbf{t'}_i = \mathbf{t}_i + \mathbf{g}_{t_i}^K \quad (5)$$

where the new template embedding with hierarchy knowledge, $\mathbf{t'}_i$, is injected into BERT replacing $\mathbf{t}_i$ in Equation 1.

4.2 Zero-bounded Multi-label Cross-entropy Loss

Since hierarchical text classification is a multi-label classification problem, previous methods (Zhou et al., 2020; Chen et al., 2021; Zhao et al., 2021) mainly regard HTC as a multiple binary classification problem and utilize the binary cross-entropy (BCE) as their loss function:

$$\mathcal{L}_{BCE} = -\sum_{i}^{C} (y_i \log(s_{y_i}) + (1 - y_i) \log(1 - s_{y_i})) \quad (6)$$

where s_{y_i} is the predicted sigmoid score of the label y_i for the input. As illustrate in Equation 6, BCE ignores the correlation between labels. In contrast, the masked language modeling is a multi-class classification task, which is optimized with the cross-entropy (CE) loss:

$$\begin{aligned} \mathcal{L}_{CE} &= -\log \frac{e^{s_{y_t}}}{\sum_{i=1}^{C} e^{s_{y_i}}} \\ &= \log(1 + \sum_{i=1, i \neq t}^{C} e^{s_{y_i} - s_{y_t}}) \end{aligned} \quad (7)$$

where y_t is the gold label for the input. As shown in Equation 7, CE forces the score of the gold label to be greater than all other labels, which directly models the label correlation.

To harmonize their objectives and bridge this *multi-label and multi-class gap*, in this paper, instead of calculating the score of each label separately, we expect the scores of all target labels are greater than all non-target labels. We use a multi-label cross-entropy (MLCE) loss (Sun et al., 2020; Su, 2020):

$$\mathcal{L}_{MLCE} = \log(1 + \sum_{y_i \in \mathcal{N}^n} \sum_{y_j \in \mathcal{N}^p} e^{s_{y_i} - s_{y_j}}) \quad (8)$$

where $\mathcal{N}^p$ and $\mathcal{N}^n$ are the target and non-target label sets of the input text.

However, Equation 8 is actually impracticable since we cannot know *a priori* the number of target labels during inference even if the positive (target) labels and negative (other) labels are separated. To fix this glitch, following Su (2020), we introduce an anchor label with a constant score 0 in MLCE and hope that the scores of the target labels and the non-target labels are all greater and less than 0 respectively. Thus, we form a zero-bounded multi-label cross-entropy (ZMLCE) loss:

$$\begin{aligned} \mathcal{L}_{ZMLCE} &= \log(1 + \sum_{y_i \in \mathcal{N}^n} \sum_{y_j \in \mathcal{N}^p} e^{s_{y_i} - s_{y_j}} \\ &+ \sum_{y_i \in \mathcal{N}^n} e^{s_{y_i} - 0} + \sum_{y_j \in \mathcal{N}^p} e^{0 - s_{y_j}}) \\ &= \log(1 + \sum_{y_i \in \mathcal{N}^n} e^{s_{y_i}}) + \log(1 + \sum_{y_j \in \mathcal{N}^p} e^{-s_{y_j}}) \end{aligned} \quad (9)$$

To be consistent with the hierarchy constraint, we adopt ZMLCE at each label hierarchy layer for the layer-wise prediction. Formally, for the m-th layer with scores predicted by $\mathbf{h}_P^m$, we add layer constraints as follows:

$$\begin{aligned} \mathcal{L}_{ZMLCE}^m =& \log(1 + \sum_{y_i \in \mathcal{N}_m^n} e^{s_{y_i}}) \\ &+ \log(1 + \sum_{y_j \in \mathcal{N}_m^p} e^{-s_{y_j}}) \end{aligned} \quad (10)$$

where $s_{y_i} = \mathbf{v}_i^T \mathbf{h}_P^m + b_{im}$ and b_{im} is a learnable bias term. $\mathcal{N}_m^p$ and $\mathcal{N}_m^n$ are the target and non-target label sets at the m-th layer for the input text respectively.

We keep the original MLM loss as BERT pre-training and the final loss $\mathcal{L}_{all}$ is the sum of ZMLCE losses at different layers and the MLM loss:

$$\mathcal{L}_{all} = \sum_{m=1}^{L} \mathcal{L}_{ZMLCE}^m + \mathcal{L}_{MLM} \quad (11)$$

We randomly mask 15% words of the text to compute the MLM loss $\mathcal{L}_{MLM}$. During inference, we select labels with scores greater than 0 as our prediction. A comparison between our method and existing prompt methods is in Appendix B.

5 Experiments

5.1 Experiment Setup

Datasets and Evaluation Metrics We experiment on Web-of-Science (WOS) (Kowsari et al., 2017), NYTimes (NYT) (Sandhaus, 2008), and RCV1-V2 (Lewis et al., 2004) datasets for analysis. The statistic details are illustrated in Table 4. We follow the data processing of previous work (Zhou et al., 2020; Chen et al., 2021) and measure the experimental results with Macro-F1 and Micro-F1.

Baselines For systematic comparisons, we introduce a variety of hierarchical text classification baselines and compare HPT with two typical prompt learning methods. 1) **TextRCNN** (Lai et al., 2015). A simple network of bidirectional GRU followed by CNN. It is a traditional text classification model adopted by HiAGM, HTCInfoMax, and HiMatch as their text encoder. 2) **BERT** (Devlin et al., 2019). A widely used pretrained language model that can serve as a text encoder. Among previous work, only HiMatch introduces BERT as text encoder so we implement other baselines with BERT replaced. 3) **HiAGM** (Zhou et al., 2020). HiAGM exploits the prior probability of label dependencies through Graph Convolution Network and applies soft attention over the text feature and label feature for the mixed feature. 4) **HTCInfoMax** (Deng et al., 2021). HTCInfoMax improves HiAGM by maximizing text-label mutual information and matching the label feature to a prior distribution. 5) **HiMatch** (Chen et al., 2021). HiMatch views the problem as a semantic matching problem and matches the relationship between the text semantics and the label semantics. 6) **HGCLR** (Wang et al., 2022). HGCLR regulates BERT representation by contrastive learning and introduces a new graph encoder.

Implement Details We implement our model using PyTorch in an end-to-end fashion. Following previous work (Chen et al., 2021), we use `bert-base-uncased` as our base architecture. We use a single layer of GAT for hierarchy injection. The batch size is set to 16. The optimizer is Adam with a learning rate of $3e^{-5}$. We train the model with the train set and evaluate on the development set after every epoch and stop training if the Macro-F1 does not increase for 6 epochs. All of the hyperparameters have not been tuned. For baseline models, we follow the hyperparameter tuning procedure in their original paper. We use a length of 8 template words for soft prompt in accordance with HPT.

5.2 Main Results

Table 1 illustrates our main results. As is shown, "HardPrompt" and "SoftPrompt" outperform the vanilla fine-tuning BERT on all 3 datasets and achieve a comparable result with the state-of-the-art method on RCV1-V2. This result shows the superiority of the prompt tuning paradigm since it adapts HTC to BERT to some extent.

By bridging the gaps between HTC and MLM, our HPT achieves new state-of-the-art results on all 3 datasets. Compared to HiMatch (Chen et al., 2021), our model introduces no extra parameter so these improvements demonstrate that HPT can better utilize the pretrained language model. Although HGCLR (Wang et al., 2022) introduces a new graph encoder, our model achieves consistent improvements on all datasets with a simple GAT. In addition, the depths of the label hierarchy for WOS, RCV1-V2, and NYT are 2, 4, and 8 respectively, which can reflect the respective difficulty of the label hierarchy. HPT outperforms both the baseline BERT and HGCLR by increasing margins on WOS, RCV1-V2, and NYT respectively, showing that hierarchy-aware prompt can better handle more difficult label hierarchy.

5.3 Ablation Study

To illustrate the effect of our proposed mechanisms, we conduct ablation studies by removing one component of our model at a time. We test on the NYT dataset in this and the following sections because

Model	WOS (Depth 2)		RCV1-V2 (Depth 4)		NYT (Depth 8)	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
TextRCNN (Zhou et al., 2020)	83.55	76.99	81.57	59.25	70.83	56.18
HiAGM (Zhou et al., 2020)	85.82	80.28	83.96	63.35	74.97	60.83
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	-	-
BERT (Wang et al., 2022)	85.63	79.07	85.65	67.02	78.24	66.08
BERT+HiAGM(Wang et al., 2022)	86.04	80.19	85.58	67.93	78.64	66.76
BERT+HTCInfoMax(Wang et al., 2022)	86.30	79.97	85.53	67.09	78.75	67.31
BERT+HiMatch (Chen et al., 2021)	86.70	81.06	86.33	68.66	-	-
HGCLR (Wang et al., 2022)	87.11	81.20	86.49	68.31	78.86	67.96
BERT+HardPrompt (Ours)	86.39	80.43	86.78	68.78	79.45	67.99
BERT+SoftPrompt (Ours)	86.57	80.75	86.53	68.34	78.95	68.21
HPT (Ours)	**87.16**	**81.93**	**87.26**	**69.53**	**80.42**	**70.42**

Table 1: F1 scores on 3 datasets. Best results are in boldface.

Ablation Models	Micro-F1	Macro-F1
HPT	**80.49**	**71.07**
r.m. hierarchy constraint	80.32	70.58
r.m. hierarchy injection	80.41	69.71
r.p. BCE loss	79.74	70.40
r.m. MLM loss	80.16	70.78
with random connection	80.12	69.42

Table 2: Performance when removing some components of HPT on the development set of NYT. *r.m.* stands for *remove*. *r.p.* stands for *replaced with*.

it has the most complicated label hierarchy and it can better demonstrate how our method reacts to the hierarchy.

After removing the hierarchy constraint, the template has only one `[PRED]` token and the model needs to recover all label words according to its hidden state. As shown in Table 2, the Micro-F1 and Macro-F1 drop slightly, which shows the effectiveness of our layer-wise prompt. By removing the hierarchy injection (i.e., remove Equation 5), the model cannot access the connectivity of the label hierarchy and drops 1.36 on Macro-F1. From this decline, we can see that the hierarchy injection is essential for the performance of labels with few instances. By incorporating an extra structural encoder, the model can learn label features from training instances from other classes based on the hierarchical dependencies between them. As a result, the hierarchy injection significantly boosts the performance of scarce classes. At last, both the performances of using BCE loss instead of ZMLCE loss (*r.p.* BCE loss) and removing MLM loss (*r.m.* MLM loss) drop, which shows it is important to bridge the gap of optimizing objectives between HTC and MLM.

To further illustrate the effectiveness of the hierarchy injection, we test our model with random connection. As a reminder, during hierarchy injection, we connect virtual nodes with according labels with the same depth. Random connection adds random connections based on that connection. For each label, it connects the label to another virtual node randomly.

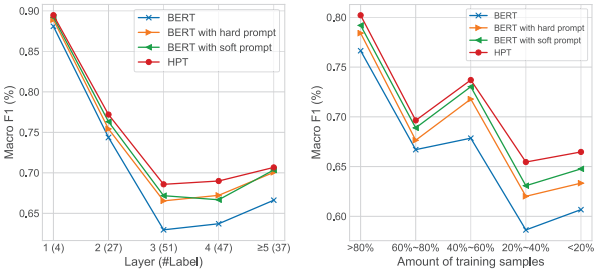
As in the last row of Table 2, the variant with random connection drops over 1% on Macro-F1 score. This result illustrates that connections that violate the label hierarchy have adverse effects. The destructiveness of a contradicting input like random connection even outweighs removing the hierarchy completely (*r.m.* hierarchy injection), reflecting that the proposed HPT indeed gains instructive information from the label hierarchy. More discussions on the connection of virtual nodes are elaborate in Appendix C. Ablation results on other datasets are in Appendix D.

5.4 Interpreting on Representation Space

In this section, we hope to intuitively show how the label hierarchy is incorporated and what the prompt has learned. The virtual label words are learned in the same space as word embedding, so they can be interpreted by their similarities with meaningful words. Therefore, we illustrate the top 8 nearest words of 2 labels in the NYT dataset, *National Hockey League* (NHL) and *News and Features* (NF). As shown in table 3, despite some

Label (different layers separated by '/')	Top 8 nearest words	
	HPT	HPT (r.m. hierarchy)
News/Sports/Hockey/ National Hockey League	[1] hockey [2] league [3] national [4] 2011 [5] 2013 [6] ##^ [7] 2012 [8] football	[1] hockey [2] national [3] league [4] 2012 [5] 2008 [6] 1996 [7] 2010 [8] 2014
Features/Theater/ News and Features	[1] features [2] . [3] and [4] the [5] theatre [6] ; [7] , [8] news	[1] . [2] features [3] and [4] the [5] , [6] ; [7] of [8] news

Table 3: Top 8 nearest words of 2 learnable virtual label words in NYT dataset.



(a) (b)

Figure 3: Macro F1 scores of label clusters on the development set of NYT. (a) Label clusters grouped by depth in the hierarchy. (b) Label clusters grouped by the number of training samples. >80% represents a cluster of top 20% labels ranking by the number of training samples. The rest clusters are arranged similarly.

meaningless words, the model indeed learns some interpretable features. For NHL, the label words of HPT consist of the semantic of *football*, which is the brother node of Hockey (the father node of NHL) in the label hierarchy. For NF, the label words of HPT consist of the semantic of *theatre*, which is the father node of NF. After removing the hierarchy knowledge (r.m. hierarchy), these semantics disappear from label words of NHL and NF. These results intuitively show that HPT incorporates the hierarchy knowledge into the pretrained language model and bridges the gap between HTC and MLM.

5.5 Results on Imbalanced Hierarchy

One of the key challenges of hierarchical text classification is the imbalanced label hierarchy. In this section, we analyze how our model resolves the issue of imbalance on the development set of NYT.

For HTC, the imbalance can be viewed from two perspectives. For one, the number of labels at different depths of the hierarchy is imbalanced. As shown in Figure 3a, medium layers (depth 3 and 4) have more labels than deep or shallow layers, where all models have poor performances. Compared with other baselines, HPT mainly boosts the performance of medium levels. For another, the instance of each label is various. Take the NYT dataset as an example, the ratio of the maximum and minimum amount of training samples of a label is over 100. In Figure 3b, we cluster labels into 5 bunches depending on their amounts of training samples. Our model largely improves the performance of labels with few training instances, showing that our method can alleviate the long-tail effect to some extent.

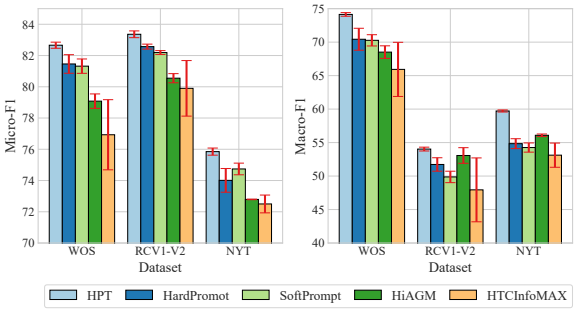


Figure 4: F1 scores on 3 mini training dataset with only 10% training instances of the full training dataset. We report the average scores with standard deviation over 3 different runs.

5.6 Results on Low Resource Setting

To further evaluate the potential of our method, we conduct experiments in low-resource settings. Since the problem is multi-label, the commonly used N-way K-shot setting is hard to define so we simply sample 10% of training data. As previous HTC works do not consider the low resource setting (LRS), we reproduce baseline models in LRS. Besides less training data, other settings follow the main experiment.

The comparison of LRS experimental results is shown in Figure 4. Among baseline methods, prompt-based models outperform non-prompt-based models on 3 datasets, which shows the advantages of prompt methods in LRS. Our model outperforms all baseline models and has better stability (lower standard deviation) on all 3 LRS datasets. Comparing with the full resource setting (FRS) (i.e., main results), the performance gap between HPT and other baselines increases on the LRS. For example, on RCV1-V2, HPT outperforms "BERT+HTCinfoMAX" 2.13 and 6.09 Macro-F1 scores in FRS and LRS, respectively, which shows the potential of our method.

6 Conclusion

In this paper, we propose a hierarchy-aware prompt tuning (HPT) method to bridge the gaps between HTC and MLM. To bridge the *hierarchy and flat gap*, HPT incorporates the label hierarchy knowledge into a virtual template and label words. To bridge the *multi-label and multi-class gap*, HPT introduces a zero-bounded multi-label cross-entropy loss to harmonize the objectives of HTC and MLM. HPT transforms HTC into a hierarchy-aware multi-label MLM task, which can better tap the potential of the pretrained language model in HTC. Extensive experiments show that our method achieves state-of-the-art performances on 3 popular HTC datasets, and is adept at handling the imbalance and low resource situations.

Limitations

Prompting methods need pretrained language models as the backbone. Our work is based on the masked language model (MLM) task but it is not a universal component of PLM. As a result, our approach is only applicable to PLMs which incorporate MLM. Despite such limited choices, compared to other HTC works which adopt PLM as a replaceable text encoder, our approach takes more advantage of PLMs by considering how they are trained. Another limitation is the constraint of maximum sequence length. Although the length limitation of PLM is extensively existed, our approach needs extra tokens for template, and that further shortens the length of input text. Even so, the experiment results indicate that our method performs better than the raw PLM so this sacrifice is worthy. Notice that the length of our template is proportional to the depth of the label hierarchy, so HPT may fail to datasets with extreme hierarchy depth.

Acknowledgements

We thank all the anonymous reviewers for their constructive feedback. The work is supported by National Natural Science Foundation of China under Grant No.62036001, PKU-Baidu Fund (No. 2020BD021) and NSFC project U19A2065.

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.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020. Hyperbolic interaction model for hierarchical multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7496–7503.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware label semantics matching network for hierarchical text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4370–4379, Online. Association for Computational Linguistics.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, pages 2778–2788.

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. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

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*, pages 257–265.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference*

on Natural Language Processing (Volume 1: Long Papers), pages 4921–4933, Online. Association for Computational Linguistics.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*.

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.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2267–2273.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. 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. 2019. 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.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212.

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

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

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.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

Jianlin Su. 2020. Extending "softmax+cross entropy" to multi-label classification problem. https://spaces.ac.cn/archives/7359.

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407.

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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7109–7119, Dublin, Ireland. Association for Computational Linguistics.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*, pages 5075–5084. PMLR.

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.

Xinyi Zhang, Jiahao Xu, Charlie Soh, and Lihui Chen. 2021. La-hcn: Label-based attention for hierarchical multi-label text classification neural network. *Expert Systems with Applications*, page 115922.

Rui Zhao, Xiao Wei, Cong Ding, and Yongqi Chen. 2021. Hierarchical multi-label text classification: Self-adaption semantic awareness network integrating text topic and label level information. In *International Conference on Knowledge Science, Engineering and Management*, pages 406–418. Springer.

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.

A Data Statistics

Dataset	$\|Y\|$	Depth	Avg($\|y_i\|$)	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 4: Data statistics. $|Y|$ is the number of classes. Depth is the maximum level of hierarchy. Avg($|y_i|$) is the average number of classes per sample.

B Example of Different Prompt Methods

We provide some detailed examples here to explain the difference between our HPT with existing prompt methods.

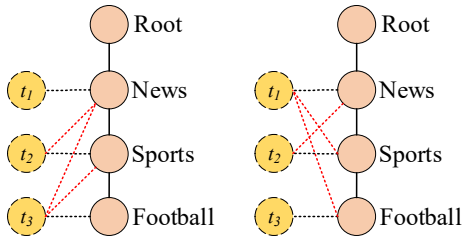
Templates of hard prompt, soft prompt, and HPT are illustrated in Table 5. **x** is the original text and `[CLS]` and `[SEP]` are special tokens of BERT. `[V1]` to `[VN]` in soft prompt are N virtual template words which are learnable embeddings, and the number N is predefined. Our method has L virtual template words. They are output embeddings of graph encoder as in Equation 5 and L is the number of hierarchy layers. Our method uses a special token `[PRED]` for multi-label prediction (Section 4.2), whereas hard and soft prompt use the same `[MASK]` token as BERT, which is proposed for single-label predictions.

Method	Template
Hard prompt	`[CLS]` **x** `[SEP]` The text is about `[MASK]` `[SEP]`
Soft prompt	`[CLS]` **x** `[SEP]` `[V1]` `[V2]` ... `[VN]` `[MASK]` `[SEP]`
HPT	`[CLS]` **x** `[SEP]` `[V1]` `[PRED]` `[V2]` `[PRED]` ... `[VL]` `[PRED]` `[SEP]`

Table 5: Example templates of hard prompt, soft prompt, and our method. **x** is the original text.

C Discussion on Different Connections of Hierarchy Injection

During hierarchy injection, we connect virtual nodes with according labels with the same depth, but this connection is not unique. Besides random connection, we further test our model with a variant. Depth increasing connects a virtual node with labels on the same and shallower layers, i.e., virtual node t_i connects with all label nodes on 1st to i-th layers. Figure 5 is an illustration of theses two connections.



(a) Depth increasing (b) Random connection

Figure 5: Two connections to aggregate node features. They add more connections (red dash line) besides the original connections (black dash line) (a) Depth increasing connects a virtual node with labels on the same and shallower layers. (b) Random connection adds random connection per node.

Ablation Models	Micro-F1	Macro-F1
HPT	**80.49**	**71.07**
r.m. hierarchy injection	80.41	69.71
with depth increasing	80.48	70.95
with random connection	80.12	69.42

Table 6: Performance of different connections of hierarchy injection on the development set of NYT. *r.m.* stands for *remove*.

As in the third row of Table 6, the variant with depth increasing behaves similarly to the original one. This observation illustrates that the impact of the connection of virtual nodes is not significant as long as it contains logical hierarchical information. Compared to random connection which violates the label hierarchy and has adverse effects, this result reflects that the proposed HPT is aware of the label hierarchy on the secondary side.

D Ablation results on WebOfScience and RCV1-V2

The hierarchy of the WOS dataset only has two layers so the structural information of WOS is weak. So, in Table 7, removing or disturbing such information has little influence.

After replacing ZMLCE loss with BCE loss, Macro-F1 decreases dramatically on all datasets.

Ablation Models	Micro-F1	Macro-F1
HPT	**87.88**	**81.68**
r.m. hierarchy constraint	87.34	81.27
r.m. hierarchy injection	87.58	81.54
r.p. BCE loss	87.17	80.78
r.m. MLM loss	87.22	81.36
with random connection	87.56	81.42

Table 7: Performance when removing some components of HPT on the development set of WOS. *r.m.* stands for *remove*. *r.p.* stands for *replaced with*.

Ablation Models	Micro-F1	Macro-F1
HPT	**88.37**	**70.12**
r.m. hierarchy constraint	87.62	69.04
r.m. hierarchy injection	87.57	68.53
r.p. BCE loss	87.79	68.12
r.m. MLM loss	87.83	69.76
with random connection	88.22	68.86

Table 8: Performance when removing some components of HPT on the development set of RCV1-V2. *r.m.* stands for *remove*. *r.p.* stands for *replaced with*.

Although BCE loss indeed can solve the multi-label problem, ZMLCE loss is a better choice theoretically and experimentally.