Exploring Label Hierarchy in a Generative Way for Hierarchical Text Classification

Wei Huang[1]*, Chen Liu[2]†, Bo Xiao[2], Yihua Zhao[2], Zhaoming Pan[2], Zhimin Zhang[2], Xinyun Yang[2], Guiquan Liu[1]
[1]University of Science and Technology of China
[2]NetEase Media Technology (Beijing) Co., Ltd.
hw001@mail.ustc.edu.cn, gqliu@ustc.edu.cn
{liuchen5,xiaobo02,hzzhaoyihua}@corp.netease.com
{panzhaoming,zhangzhimin,yangxinyun}@corp.netease.com

Abstract

Hierarchical Text Classification (HTC), which aims to predict text labels organized in hierarchical space, is a significant task lacking in investigation in natural language processing. Existing methods usually encode the entire hierarchical structure and fail to construct a robust label-dependent model, making it hard to make accurate predictions on sparse lower-level labels and achieving low Macro-F1. In this paper, we explore the level dependency and path dependency of the label hierarchy in a generative way for building the knowledge of upper-level labels of current path into lower-level ones, and thus propose a novel PAAM-HiA-T5 model for HTC: a hierarchy-aware T5 model with path-adaptive attention mechanism. Specifically, we generate a multi-level sequential label structure to exploit hierarchical dependency across different levels with Breadth-First Search (BFS) and T5 model. To further improve label dependency prediction within each path, we then propose an original path-adaptive attention mechanism (PAAM) to lead the model to adaptively focus on the path where the currently generated label is located, shielding the noise from other paths. Comprehensive experiments on three benchmark datasets show that PAAM-HiA-T5 greatly outperforms all state-of-the-art HTC approaches especially in Macro-F1.

1 Introduction

Hierarchical text classification (HTC), where text labels are predicted within a hierarchical structure, is a challenging task that has not yet received due attention within the field of multi-label classification. HTC methods have been extensively applied in industry domains, e.g., news article classification (Sandhaus, 2008), product classification in E-commerce (Yu et al., 2018), bidding strategy in paid search marketing (Agrawal et al., 2013).

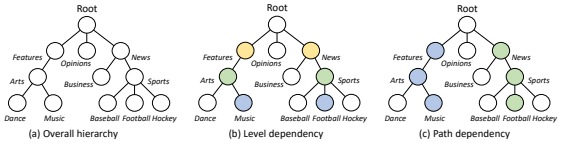

Figure 1: (a): the static labeling process is uniform and simultaneous for all labels in the label hierarchy. (b): the dynamic labeling process where the lower-level labels depend on the upper-level labels. (c): the dynamic labeling process focuses on ancestor labels already generated on the current path.

In HTC tasks, labels at lower-level are inevitably sparse due to the hierarchical structure. Many studies (Barbedo and Lopes, 2006; Xiao et al., 2007; Johnson and Zhang, 2015) completely or partially neglect such hierarchical structure and fail to accurately predict those lower-level labels, achieving low Macro-F1 score. Existing studies (Peng et al., 2021; Wu et al., 2019) have proved that introducing structure information can boost the predictive power on low-level labels and thus improve the overall task performance. A number of studies (Cesa-Bianchi et al., 2006; Shimura et al., 2018; Wehrmann et al., 2018; Banerjee et al., 2019) propose to construct multi-level classifiers that are trained independently and predicted sequentially, where only local maximum is achieved and propagation of error negatively impacts model prediction. Some studies design end-to-end models that introduce various strategies (such as Tree-LSTM/GCN (Zhou et al., 2020), label semantics matching network (Chen et al., 2021), graph-CNN (Peng et al., 2018) and hierarchical fine-tuning based CNN (Shimura et al., 2018)) to encode the overall hierarchy information (as depicted in Figure 1 (a)), where label dependency across different levels (as depicted in Figure 1 (b)) is not captured in a more principled way and unnecessary noises are introduced. Such models tend to predict all labels simultaneously and independently with sigmoid function, and they could cause serious *label inconsistencies*

*Contribution during internship at NetEase Inc.
†Corresponding Author

Proceedings of the 29th International Conference on Computational Linguistics, pages 1116–1127
October 12–17, 2022.

(One label is predicted positive but its ancestors are not. For example, in Figure 1 (b), the "Football" is predicted while the "Sports" is not.) and require post-processing to rectify these contradictions (Mao et al., 2019). Although one recent study (Mao et al., 2019) develops label-dependent models with reinforcement learning, it still fails to address label dependency within each path (as depicted in Figure 1 (c)) and fails to fully integrate labels and text information.

This paper seeks to close the gap by proposing the PAAM-HiA-T5. We are not only the first to capture lower-level label dependency on upper-level ones with generation model, but also the first to accurately explore the level dependency within each path. In each step of prediction phase, our model generates next label based on the text sequence and labels previously generated on current path. As illustrated in Figure 1(c), our model sequentially predicts "Features", "News", "Arts", "Sports", "Music", "Football" labels. In the process where label "Football" is generated, our model pays more attention on "News" and "Sports" labels on its own path instead of "Features" and "Arts" labels on another path. Therefore, our model is less likely to cause label inconsistency. For example, when labels "News" and "Sports" are known to the model, it is easier to predict the label "Football" as Figure 1 (b) shows.

Our PAAM-HiA-T5 model follows a two-step design.

Hierarchy-aware T5 (HiA-T5), a variant of T5 that is fully aware of the level dependency in a multi-level sequential generative manner. Inspired by the idea that conventional classification routines often follow the order that from coarse-grained to fine-grained to predict labels, we firstly use Breadth-First Search (BFS) to flatten hierarchical labels into multi-level sequential label structure, transforming the hierarchy to sequence. T5 model is applied to map the text sequence to label sequence, where the text sequence and upper-level labels generated earlier are then integrated in order to determine the next label.

Path-adaptive attention mechanism (PAAM), a mechanism to exploit the label correlation within each path and shield the noise from other paths. Through the PAAM, the model can adaptively obtain a more reasonable attention distribution belonging to the path where the currently generated label is located. PAAM is implemented by means of a regularization method.

This study makes the following major contributions:

- We propose a novel HiA-T5, a multi-level sequential label generative model to exploit label dependency across different levels. The mapping relationship between text sequence and label sequence is examined in each step of prediction.
- We propose an original PAAM to lead the model to adaptively focus on the path where the currently generated label is located, shielding the noise from other paths to further improve prediction performance.
- Experiments on various datasets show that our proposed PAAM-HiA-T5 achieves significantly and consistently better performance than state-of-the-art models. Experimental analysis shows that PAAM-HiA-T5 is especially benefical to those lower-level long-tailed labels. And our model can obtain better label consistency.

2 Related Work

Hierarchical text classification (HTC) is a critical task with numerous applications (Qu et al., 2012; Agrawal et al., 2013; Zhang et al., 2019; Peng et al., 2016). By methods of hierarchical information modeling, HTC approaches can be categorized into flat, local and global approaches (Silla and Freitas, 2011).

Flat approaches (Hayete and Bienkowska, 2005; Barbedo and Lopes, 2006; Xiao et al., 2007; Johnson and Zhang, 2015) completely or partially ignore the label hierarchy and each label is independently predicted. Some of them simply ignore the invaluable hierarchical information and achieve poor performance. Some others predict leaf nodes first and then mechanically add their ancestor labels, which is only applicable where different paths in the label hierarchy share the same length.

Local approaches (Koller and Sahami, 1997; Cesa-Bianchi et al., 2006; Shimura et al., 2018; Wehrmann et al., 2018; Banerjee et al., 2019) construct multiple local classifiers so that the misclassification at a certain level is propagated downwards the hierarchy, easily leading to the exposure of bias (Silla and Freitas, 2011). Specifically, Peng et al. (2018) proposes deep graph convolutional

neural networks with hierarchical regularization. Wehrmann et al. (2018) utilizes a multi-label neural network architecture with local and global optimization. To address the lower-level labels sparsity problem, Shimura et al. (2018) takes advantage of a CNN-based model with the fine-tuning method. Banerjee et al. (2019) proposes to transfer the parameters of parent classifiers to initialize child classifiers for HTC task.

Global approaches (Gopal and Yang, 2013; Mao et al., 2019; Wu et al., 2019; Zhou et al., 2020; Peng et al., 2021), where the entire structural information is encoded and all labels are simultaneously predicted, has become recent mainstream due to its better performance. Mao et al. (2019) handles HTC task with reinforcement-learning-based label assignment method. Wu et al. (2019) uses meta-learning to model the label interaction for multi-label classification. Zhou et al. (2020) utilizes the Bi-TreeLSTM and GCN to model hierarchical relationship and makes flat predictions for hierarchical labels. Peng et al. (2021) combines CNN, RNN, GCN, and CapsNet to model hierarchical labels. Chen et al. (2021) formulate HTC as a semantic matching problem to mine the text-label semantics relationship. Although recent researchers have managed to introduce hierarchical information in different fashions, most of them still regard flat multi-label classification as the backbone of HTC where all labels are predicted simultaneously and independently. Their exploitation of hierarchical structure is far from sufficient.

3 Problem Definition

For HTC, we define the overall label hierarchy as a tree-like structure, denoted by $T = (L, E)$, where $L = \{l_1, l_2, \ldots, l_K\}$ refers to the set of all label nodes in the corpus and K is the total number of them. E refers to the set of edges indicating the nodes' parent-child relations. Formally, we denote text objects as $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$ and their labels as $\mathcal{L} = \{L_1, L_2, \ldots, L_N\}$.

Each text object is represented by a text sequence $X_i = [x_1, x_2, \ldots, x_J]$, where x_j is a word and J is the number of words in the text object. Meanwhile, each text object X_i is mapped to a original label set $L_i = \{l_1, l_2, \ldots, l_k, 1 \leq k < K\}$ that contains multiple labels. We then define a set of special symbols $S = \{\_, /, EOS\}$ to identify special hierarchical relationships in the hierarchy.

All labels L in the corpus constitute the overall label hierarchy T. The original label set $L_i = \{l_1, l_2, \ldots, l_k, 0 \leq k < K\}$ of any text object X_i constitute an partial label hierarchy T_i and $T_i \subset T$. We aim to train a model to predict corresponding label set L_i for each text object X_i, where the label set L_i is constrained by the hierarchy T_i.

4 Background

The T5 model consists of an encoder-decoder architecture, which mainly includes the Multi-head Attention Mechanism, the Feed-Forward Network and so on (Raffel et al., 2020), as depicted in the Figure 2.

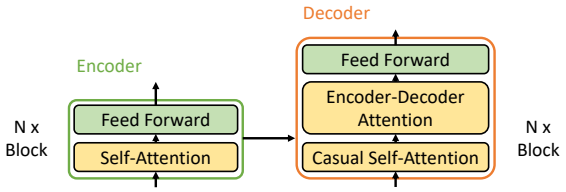

Figure 2: Structure of T5. Following each Multi-head Attention sublayer and Feed-Forward sublayer, there are a series of dropout, residual connection and layer normalization. These parts are omitted in the figure and the following formulas for simplicity's sake.

Attention mechanism $\mathrm{Attn}(\cdot)$ is calculated as:

$$\begin{aligned} \mathrm{Attn}(Q, K, V) &= \mathrm{Score}(Q, K)V \\ \mathrm{Score}(Q, K) &= \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}}) \end{aligned} \tag{1}$$

where $Q, K, V \in \mathbb{R}^{n \times d_{model}}$ and the length of sequence is n. The output attention score matrix of $\mathrm{Score}(Q, K)$ is denoted as $Score \in \mathbb{R}^{n \times n}$.

Multi-head attention independently executes attention mechanism of H heads and then concatenate their results, and it is denoted as $\mathrm{MH}(\cdot)$. The Feed-Forward Network consists of two linear transformations with a nonlinear activation function in between, and it is represented as $\mathrm{FFN}(\cdot)$.

T5 encoder is composed of a stack of "encoder blocks" and we define the number of blocks as B. Each block contains a self-attention sublayer and a feed-forward sublayer. The input sequence of encoder is mapped to the embedding $Q_{encoder}, K_{encoder}, V_{encoder} \in \mathbb{R}^{n \times d_{model}}$, which are then passed into the encoder. The output of encoder is denoted as $O_{encoder}$.

$$\begin{aligned} &\mathrm{Block}_{\mathrm{Encoder}}(Q_{encoder}, K_{encoder}, V_{encoder}) \\ &= \mathrm{FFN}(\mathrm{MH}(Q_{encoder}, K_{encoder}, V_{encoder})) \\ &\mathrm{Encoder}(Q_{encoder}, K_{encoder}, V_{encoder}) \\ &= \mathrm{stack}(\mathrm{Block}_{\mathrm{Encoder}}(Q_{encoder}, K_{encoder}, V_{encoder})) \end{aligned} \tag{2}$$

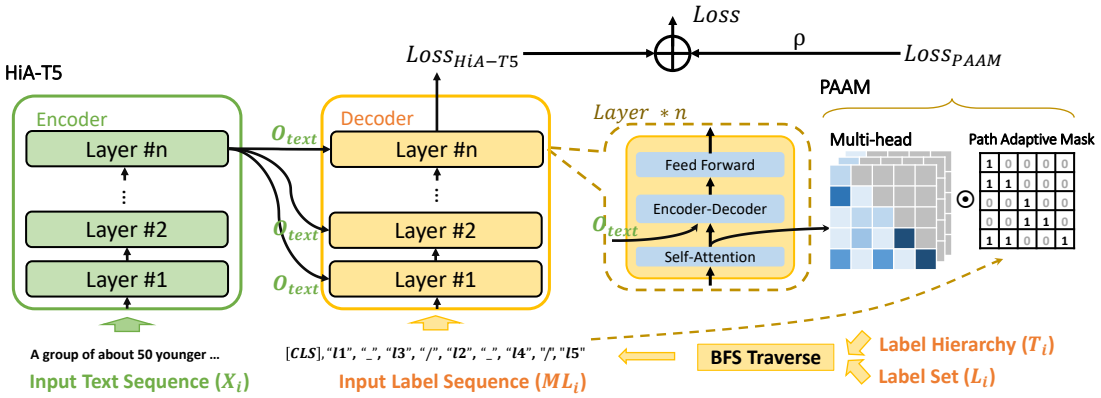

Figure 3: The overall structure of PAAM-HiA-T5. PAAM-HiA-T5 consists of a HiA-T5 and a PAAM. The dataflows of one decoder layer are illustrated in the yellow dashed box.

The structure of the decoder looks similar to that of the encoder, except that it has an additional encoder-decoder attention sublayer that attends to the output of the encoder stack, following each casual self-attention sublayer. The causal attention mechanism of decoder only permits the model attend to past outputs. In the end, we obtain the decoder output denoted as $O_{decoder}$.

$$\begin{aligned}&\text{Block}_{\text{Decoder}}(Q_{decoder}, K_{decoder}, V_{decoder}, O_{encoder})\\&= \text{FFN}(\text{MH}(\text{MH}(Q_{decoder}, K_{decoder}, V_{decoder}), O_{encoder}, O_{encoder}))\\&\text{Decoder}(Q_{decoder}, K_{decoder}, V_{decoder}, O_{encoder})\\&= \text{stack}(\text{Block}_{\text{Decoder}}(Q_{decoder}, K_{decoder}, V_{decoder}, O_{encoder}))\end{aligned} \quad (3)$$

5 Hierarchy-Aware T5 with Path-Adaptive Attention Mechanism

As depicted in Figure 3, we propose a PAAM-HiA-T5 model for HTC: a **Hi**erarchy-**A**ware **T5** model with **P**ath-**A**daptive **A**ttention **M**echanism. PAAM-HiA-T5 consists of the HiA-T5 for level-dependent label generation and the PAAM for path-specific label generation.

5.1 Hierarchy-Aware T5

Level-dependent HiA-T5 The major shortcoming of previous HTC methods is the inadequate application of hierarchy information. In contrast, HiA-T5 exploits label dependency across different levels of the hierarchy with Breadth-First Search (BFS) and multi-head attention mechanism.

HiA-T5 firstly explore the label hierarchy T_i with Breadth-First Search (Cormen et al., 2001) to flatten the label set $L_i = \{l_1, l_2, l_3, l_4, l_5\}$ into multi-level sequential label $ML_i = [CLS, l_1, \_, l_3, /, l_2, \_, l_4, /, l_5]$, transforming the hierarchy to multi-level label sequence, as illustrated in Figure 4 (a). In this process, '_' between labels denotes intra-level relationship, while '/' signifies inter-level relationship.

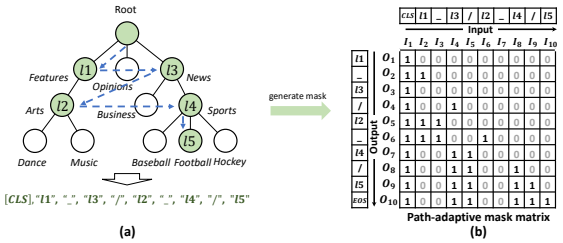

Figure 4: (a): A text with the content "David Beckham attends art exhibition launch during Paris Fashion Week" is tagged with "News", "Sports", "Football", "Features" and "Arts". And its label hierarchical structure is explored in Breadth-First Search (blue dash line). (b): path-adaptive mask matrix makes the ith output element use current input element and all its ancestors.

On one hand, the text sequence $X_i = [x_1, x_2, \ldots, x_J]$ is mapped to embedding sequence $Q_{text}, K_{text}, V_{text} \in \mathbb{R}^{n_t \times d_{model}}$ of length n_t, which are then passed into T5 encoder:

$$O_{text} = \text{Encoder}(Q_{text}, K_{text}, V_{text}) \quad (4)$$

The output encoder representation for semantic features of varied granularities is O_{text}.

On the other hand, the multi-level label sequence $ML_i = [CLS, l_1, \_, l_3, /, l_2, \_, l_4, /, l_5]$ is mapped to embeddings sequence $Q_{label}, K_{label}, V_{label} \in \mathbb{R}^{n_l \times d_{model}}$ of length n_l, which are passed into T5 decoder together:

$$O_{hierarchy} = \text{Decoder}(Q_{label}, K_{label}, V_{label}, O_{text}) \quad (5)$$

Specifically, HiA-T5 fully explores the label dependency across different levels through the self-attention mechanism. With the help of the intra-level separator '_' and the inter-level separator '/', the causal decoder self-attention mechanism fully excavates the intra-level parallel and mutually ex-

clusive relationship, as well as the inter-level dependent and appurtenant relationship. The output representation of the decoder causal self-attention mechanism incorporating level dependency information is $A_{label} = \mathrm{MH}(Q_{label}, K_{label}, V_{label})$.

So far, we have obtained the text representation O_{text} highlighting the semantic features of texts with different granularities and the label representation A_{label} incorporating label dependency across different levels. The output representation of the encoder-decoder attention mechanism integrating these two is $A_{cross} = \mathrm{MH}(A_{label}, O_{text}, O_{text})$, which is a sufficient crossover information for following prediction.

Loss of HiA-T5 We have obtained the final decoder block output $O_{hierarchy}$ of HiA-T5, which fully integrates the label hierarchy information and the text semantic information of different granularities. Then $O_{hierarchy}$ is passed into a fully connected layer with a softmax output, which is also the final result of HiA-T5 denoted as $Pred$. $Pred$ is the result of n_l timesteps and $Pred \in \mathbb{R}^{n_l \times K}$.

$$Pred = \mathrm{softmax}(O_{hierarchy} W_3 + b_3) \quad (6)$$

where $W_3 \in \mathbb{R}^{d_{model} \times K}, b_3 \in \mathbb{R}^K$. In addition, any multi-level label sequence ML_i is transformed into $Truth \in \mathbb{R}^{n_l \times K}$, which is composed of one-hot vectors corresponding to all labels. Therefore, the cross-entropy loss of HIA-T5 is expressed as:

$$\mathcal{L}_{HiA-T5} = \mathrm{crossentropy}(Truth, Pred) \quad (7)$$

5.2 Path-Adaptive Attention Mechanism

PAAM is a mechanism that can lead the model to adaptively focus on the path where the currently generated label is located, shielding the noise from other paths. It is a regularization method designed in the training phase to encourage the model to pay more attention to ancestor labels on current path while penalizing those on other paths. We first obtain the path-adaptive mask matrix containing hierarchy information. Then path-adaptive attention loss is obtained according to operations on the path-adaptive mask matrix and the causal attention score matrix.

Path-Adaptive Mask Matrix Now the text sequence $X_i = [x_1, x_2, \ldots, x_J]$ is taken as input, which is fed into HiA-T5 for training, and its corresponding multi-level label sequence ML_i is taken as output.

According to the T5 structure of Figure 2, the sequence ML_i is first passed into causal attention sub-layer of decoder. Within this sub-layer, and according to formula (1), we get causal attention score matrix $Score$ of one head corresponding to the sequence ML_i, as depicted in Figure 5 (a).

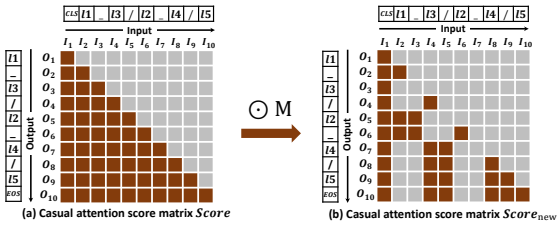

Figure 5: (a): Causal attention score matrix $Score$. The input and output of the causal self-attention mechanism are denoted as I and O respectively. $Score = \{s_{i,j}\} \in \mathbb{R}^{n_l \times n_l}$. Each element $s_{i,j}$ at row i and column j represents the weight at which the self-attention mechanism attends to input element j at output timestep i. The gray cell indicates the corresponding attention score $s_{i,j} = 0$. (b): Element-wise product result of $Score$ and M.

Then we define the path-adaptive mask matrix M, which can mask different parts of the label sequence at different decoding timestep i. The matrix M is obtained from the hierarchical structure of the label sequence corresponding to each text object. Specifically, the shape of matrix M is same as that of attention score matrix $Score$. Mask matrix $M = \{m_{i,j}\} \in \mathbb{R}^{n_l \times n_l}$ is also a lower triangular matrix, and it is only composed of 0 or 1 as shown below.

$$M = \begin{bmatrix} m_{1,1} & & & & 0 \\ m_{2,1} & m_{2,2} & & & \\ \vdots & \vdots & \ddots & \ddots & \\ m_{n_l,1} & m_{n_l,2} & \cdots & m_{n_l,n_l-1} & m_{n_l,n_l} \end{bmatrix} \quad (8)$$

O_i represents the input of the attention mechanism at the timestep i, and $O_i \in L \cup S$. If $O_i \in L$, we define $ancestor(O_i)$ as label O_i's ancestor labels and the special symbol immediately following them.

Then we define the following formula to fill the matrix M based on the parent-child relationship contained in each path of label hierarchy. The output timestep i and input timestep j correspond to the i-th row and j-th column of the matrix M. The element $m_{i,j}$ is determined by both the output O_i and the input I_j, and the formula is as follows:

$$m_{i,j} = \begin{cases} 1 & \{O_i \in L, I_j \in ancestor(O_i)\} \\ & \cup \{O_i \in S, I_j \in ancestor(O_{i-1}) \ or \ I_j = O_{i-1}\} \\ & \cup \{O_i \in L \cup S, I_j = CLS\} \\ 0 & else \end{cases} \quad (9)$$

Path-Adaptive Attention Loss With text sequence and labels previously generated in hand, we now introduce regularization and apply the path-adaptive dynamic mask matrix M, such that HiA-T5 decoder learns the weight of the attention matrix and pays more attention on the label's current path.

Having obtained the multi-level label sequence ML_i of a certain training sample, we use it as the input of causal self-attention of HiA-T5's decoder. According to the definition above, we get its path-adaptive mask matrix M, as depicted in Figure 4 (b). Furthermore, we get path-adaptive attention score matrix $Score_{path}$ as depicted in Figure 5 (b) by multiplying attention score matrix $Score$ and the path-adaptive mask matrix M element-wise:

$$Score_{path} = Score \odot M = softmax(\frac{QK^T}{\sqrt{d_k}}) \odot M \quad (10)$$

We define C as the index set of $ancestor(I_i)$. At any decoding timestep i, our goal is to make the sum of the attention scores $\sum_{j \in C} s_{i,j}$ of current path's labels as close to 1 as possible. Corresponding to attention scores matrix $Score_{path}$ of decoder's causal attention, that is, to make the sum of elements of each row in the matrix close to 1 as much as possible. According to the definition section, suppose $Score$ is the causal attention score matrix corresponding to the h-th head of b-th decoder "blocks", where $1 \leq h \leq H, 1 \leq b \leq B$. The path-adaptive attention loss is defined as:

$$\mathcal{L}_{PAAM} = \sum_{b=1}^{B} (\frac{\sum_{h=1}^{H} (\sum_{i=1}^{n_l} (1 - \sum_{j \in C} s_{i,j}))}{H}) \quad (11)$$

Therefore, the path-adaptive attention loss is added to the loss of HiA-T5 as total loss for training. The total loss function $Loss$ is obtained as below, where ρ is the coefficient of path-adaptive attention loss item.

$$\mathcal{L} = \mathcal{L}_{HiA-T5} + \rho \mathcal{L}_{PAAM} \quad (12)$$

6 Experiments

6.1 Experiment Setup

Datasets We conduct experiments on three public datasets, including RCV1-V2 (Lewis et al., 2004), NYTimes(NYT) (Sandhaus, 2008) and Web-of-Science(WOS) (Kowsari et al., 2017). RCV1-V2 is an English news categorization dataset and NYT is a news dataset from the New York Times in America. WOS is about scientific literature categorization. Labels of these datasets are organized into a tree-like structure. Relevant information of datasets is summarized in Table 1 and Table 2.

Dataset	$\|L\|$	Depth	Avg($\|L_i\|$)	Max($\|L_i\|$)	Train	Val	Test
RCV1	103	4	3.24	17	20833	2316	781265
NYT	166	8	7.6	38	23345	5834	7292
WOS	141	2	2.0	2	30070	7518	9397

Table 1: Statistical analysis of datasets: $|L|$ is the number of all labels in the hierarchy. Depth denotes the maximum level of the label hierarchy. Avg($|L_i|$) and Max($|L_i|$) denote average and maximum number of labels in each sample.

Dataset	level1	level2	level3	level4	level5	level6	level7	level8
RCV1	236334	20523	11850	23211	-	-	-	-
NYT	15161	2923	1160	842	1066	925	992	1460
WOS	6712	351	-	-	-	-	-	-

Table 2: Statistics of the average number of each label's occurrence at each level: $level_i$ denotes the level in the label hierarchy. In general, the labels of most samples are screwed towards upper levels and lower-level labels are more sparse.

We split RCV1-V2 in the benchmark split manner and take a small portion of the training set as validation set. For NYT and WOS, we randomly split data into training, validation and test sets.

Evaluation Metrics We use standard evaluation metrics, including **Micro-F1** and **Macro-F1** (Gopal and Yang, 2013; Peng et al., 2018; Huang et al., 2019), to measure the performance of all HTC methods. Micro-F1 equally weights all samples, while Macro-F1 gives equal weight to each label. As such, Micro-F1 gives more weight to frequent labels, while Macro-F1 equally weights all labels and is more sensitive to lower-level sparse labels which are shown in Table 2.

Experimental Settings The backbone pre-trained model we adopt is T5-base (Raffel et al., 2020) containing about 220M parameters. Tokenizer from T5 is utilized to preprocess the text. For PAAM-HiA-T5, the maximum sequence length of encoder is set as 300 for all datasets, and the maximum sequence length of decoder for RCV1, NYT and WOS are respectively set as 90, 120 and 20. When the model is trained, Adam optimizer is employed in a batch size of 10 with learning rate of 5e-4. The search range of coefficient ρ is {0.1,1,10,100,200}, and we set it to 100, 10, and 100 for RCV1, NYT and WOS respectively according to validation results. In the inference phase, greedy search is adopted. We set random seeds before experiments for the reproducibility of results, and the results reported in this paper are from the

Models	Micro-F1	Macro-F1
Flat Models		
HAN (Mao et al., 2019)	75.30	40.60
TextCNN (Mao et al., 2019)	76.60	43.00
TextRCNN (Zhou et al., 2020)	81.57	59.25
Local Models		
HR-DGCNN-3 (Peng et al., 2018)	76.18	43.34
HFT(M) (Shimura et al., 2018)	80.29	51.40
Htrans (Banerjee et al., 2019)	80.51	58.49
HMCN (Mao et al., 2019)	80.80	54.60
Global Models		
SGM (Zhou et al., 2020)	77.30	47.49
HE-AGCRCNN (Peng et al., 2021)	77.80	51.30
HiLAP-RL (Mao et al., 2019)	83.30	60.10
HiAGM (Zhou et al., 2020)	83.96	63.35
HiMatch (Chen et al., 2021)	84.73	64.11
Pretrained Language Models		
BERT	86.26	67.35
T5	86.14	67.39
BERT+HiAGM[1]	86.12	68.08
BERT+HiMatch (Chen et al., 2021)	86.33	68.66
PAAM-HiA-T5	**87.22**	**70.02**

Table 3: Performance comparison on RCV1-V2. The results of HAN (Yang et al., 2016), TextCNN (Kim, 2014) and HMCN (Wehrmann et al., 2018) are reported by Mao et al. (2019). Zhou et al. (2020) reports the results of TextRCNN (Lai et al., 2015) and SGM (Yang et al., 2018).

average of 3 random runs of the model.

6.2 Performance Comparison

The experimental comparison between PAAM-HiA-T5 and the state-of-the-art HTC methods are shown in Table 3 and 4, and our model outperforms all SOTA results of flat, local and global methods, both on Micro-F1 and Macro-F1. This demonstrates the strong power of PAAM-HiA-T5 in solving HTC by better mining hierarchical structure information. The level-dependency modeling and the path-adaptive attention mechanism bring significant improvement. HiAGM and HiMatch are effective baselines because they achieved the latest SOTA results in HTC. Our model greatly surpasses them on both metrics especially on Macro-F1. In general, the greater improvement on Macro-F1 shows that our model has greater capability in predicting sparse lower-level labels. In fact, it can be shown from Table 2 that sample labels become more sparse as level grows. Due to insufficient training, the lower-level label prediction becomes difficult. But our model utilizes the knowledge of upper-level labels in predicting lower-level ones by modeling level dependency and path dependency, and this explains the reason why our model achieves greater boost in Macro-F1 and has greater capability in predicting sparse lower-level labels.

Pre-trained language models are effective methods, which can often be combined with the existing model structure to improve the performance of specific tasks. BERT+HiMatch and BERT+HiAGM denote that HiMatch and HiAGM are respectively equiped with a pre-trained BERT (Kenton and Toutanova, 2019) compatible with their structures. The model sizes of BERT+HiMatch and BERT+HiAGM are in the same order of magnitude as that of PAAM-HiA-T5. Our model can still significantly outperform them, which shows the powerful capabilities of it (see Appendix C for more analysis on this). For a more detailed discussion about computational complexity please refer to the Appendix A.

Model	NYT		WOS	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Flat Models				
TextRNN (Zhou et al., 2020)	70.29	53.06	77.94	69.65
TextCNN (Zhou et al., 2020)	70.11	56.84	82.00	76.18
TextRCNN (Zhou et al., 2020)	70.83	56.18	83.55	76.99
Local & Global Models				
HMCN (Mao et al., 2019)	72.2	47.4	–	–
HiAGM (Zhou et al., 2020)	74.97	60.83	85.82	80.28
HiMatch (Chen et al., 2021)	74.62	59.28	86.20	80.53
Pretrained Language Models				
BERT+HiMatch[2] (Chen et al., 2021)	76.79	63.89	86.70	81.06
PAAM-HiA-T5	**77.52**	**65.97**	**90.36**	**81.64**

Table 4: Performance comparison on the NYT and WOS datasets. We mainly compare the best performing flat, local, global and pre-trained models on RCV1-V2. The results of TextRNN (Liu et al., 2016), TextCNN (Kim, 2014) and TextRCNN (Lai et al., 2015) on NYT and WOS are reported by Zhou et al. (2020).

6.3 Performance Analysis

Method	Micro-F1	Macro-F1
T5	86.14	67.39
HiA-T5	86.67	69.09
PAAM-HiA-T5	87.22	70.02

Table 5: Ablation study of PAAM-HiA-T5 on RCV1-V2. Note that original T5 neither model the hierarchical structure information nor capture the hierarchial dependencies. It takes HTC as a generic multi-label classification task to generate unordered label sets corresponding to the text.

[1]The results of BERT+HiAGM on RCV1-V2 are implemented upon the released projects of HiAGM (https://github.com/Alibaba-NLP/HiAGM) and the BERT with mutli-label settings. We follow the MIT License.

[2]The results of HiMatch and BERT+HiMatch on NYT is reproduced upon the released project of HiMatch (https://github.com/RuiBai1999/HiMatch) and the BERT with mutli-label settings. We follow the MIT License.

Ablation Study and Analysis on Level Dependency Modeling The performance comparison of HiA-T5 and the original T5 is shown in Table 5. It is evident that HiA-T5 greatly outperforms the T5 both in Micro-F1 and Macro-F1. This result illustrates the effectiveness of capturing level dependency by introducing upper-level label knowledge to assist lower-level label prediction. Compared with T5, HiA-T5 boosts Macro-F1 by 1.70% and achieves substantial 0.53% improvement in Micro-F1. Greater boost in Macro-F1 demonstrates HiA-T5 is especially beneficial to lower-level long-tailed labels by introducing level dependency modeling.

Ablation Study and Analysis on Path-adaptive Attention Mechanism The performance comparison of PAAM-HiA-T5 and HiA-T5 is also shown in Table 5. PAAM-HiA-T5 greatly increases Micro-F1 and Macro-F1 especially in Macro-F1 compared with HiA-T5. This indicates that PAAM significantly improves the performance of HiA-T5 in more challenging multi-path scenarios by capturing precise path dependency.

As shown in Figure 6, the heat map of the causal self-attention score in PAAM-HiA-T5's encoder proves the effectiveness of PAAM, where the attention score is mainly distributed on the path of the label current being decoded.

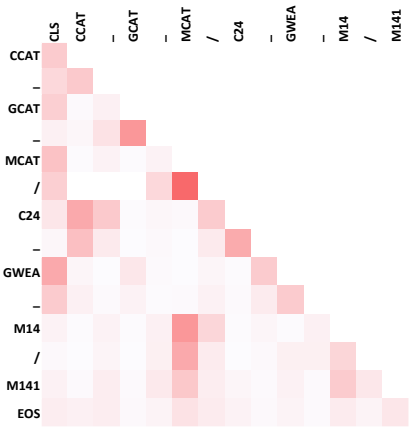

Figure 6: Heatmap of causal self-attention score for a random sample. Note that symbols instead of original labels are used and the score of each label is the average of its tokens' attention score for ease of display.

Performance Analysis on Label Granularity To find the origin of performance improvement, we analyze performance on label granularity based on different levels for T5, HiA-T5 and PAAM-HiA-5 on RCV1-V2. Figure 7 shows the level-based Macro-F1 of models and the absolute Macro-F1 differences among models. In general, our mechanism and strategy brings performance improvement on all levels, especially on lower levels.

In addition, the gap between HiA-T5 and T5 gets bigger as level deepens, and the phenomenon between HiA-T5 and PAAM-HiA-T5 is consistent. This illustrates that as the level grows, label prediction becomes more and more difficult, and the introduction of upper-level label knowledge by leveraging level dependency modeling and path-adaptive attention mechanism becomes more and more valuable. Specifically, the Macro-F1 of second and third levels for T5 are relatively low because there are some long-tailed labels among lower levels, but HiA-T5 and PAAM-HiA-T5 greatly enhance them. More performance comparison on label granularity with SOTA methods are provided in Appendix B.

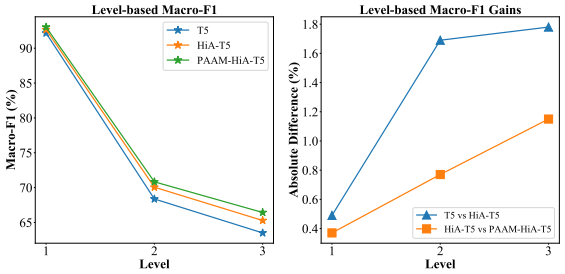

Figure 7: Ablation analysis based on different levels.

Analysis of Label Consistency Label inconsistency is a serious problem in many HTC approaches, due to the fact that they focus on flat multi-label classification and make independent predictions for all labels. It is worth mentioning that PAAM-HiA-T5 has outstanding classification performance while maintaining an ultra-low label inconsistency rate of 0.31%, as shown in Table 6. This is because our model fully leverages the constraints of upper-level labels generated earlier to predict the most accurate lower-level labels.

TextCNN	HMCN	HiAGM	HiMatch	BERT+HiAGM	BERT+HiMatch	**PAAM-HiA-T5**
3.74%	3.84%	1.35%	1.33%	1.52%	1.14%	**0.31%**

Table 6: Comparison of label inconsistency on RCV1-V2. We calculate the label inconsistency as the ratio of predictions with inconsistent labels. The results of TextCNN and HMCN are reported in Mao et al. (2019).

7 Conclusion

For HTC task, we explicitly define the concepts of "level dependency" and "path dependency" for the first time. Furthermore, in order to build the

knowledge of upper-level labels into lower-level ones in HTC task, we devise an innovative PAAM-HiA-T5 methodology by exhaustively exploring level dependency and path dependency of hierarchy in a generative manner. Comprehensive experiments on three benchmark datasets show that our model greatly outperforms all state-of-the-art HTC approaches especially in Macro-F1.

References

Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 13–24, New York, NY, USA. Association for Computing Machinery.

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.

Jayme Garcia sArnal Barbedo and Amauri Lopes. 2006. Automatic genre classification of musical signals. *EURASIP Journal on Advances in Signal Processing*, 2007:1–12.

Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. 2006. Hierarchical classification: Combining bayes with svm. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 177–184, New York, NY, USA. Association for Computing Machinery.

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.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2001. Introduction to algorithms second edition. *The Knuth-Morris-Pratt Algorithm*.

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.

Boris Hayete and Jadwiga R Bienkowska. 2005. Gotrees: predicting go associations from protein domain composition using decision trees. In *Biocomputing 2005*, pages 127–138. World Scientific.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1051–1060, New York, NY, USA. Association for Computing Machinery.

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.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Daphne Koller and Mehran Sahami. 1997. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, page 170–178, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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 *Twenty-ninth AAAI conference on artificial intelligence*.

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.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

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.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1063–1072, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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.

Shengwen Peng, Ronghui You, Hongning Wang, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2016. Deepmesh: deep semantic representation for improving large-scale mesh indexing. *Bioinformatics*, 32(12):i70–i79.

Bo Qu, Gao Cong, Cuiping Li, Aixin Sun, and Hong Chen. 2012. An evaluation of classification models for question topic categorization. *Journal of the American Society for Information Science and Technology*, 63(5):889–903.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Evan Sandhaus. 2008. The New York Times Annotated Corpus.

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.

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.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 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.

Zhongzhe Xiao, Emmanuel Dellandrea, Weibei Dou, and Liming Chen. 2007. Hierarchical classification of emotional speech. *IEEE Transactions on Multimedia*, 37.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Wenhu Yu, Zhiqiang Sun, Haifeng Liu, Zhipeng Li, and Zhitong Zheng. 2018. Multi-level deep learning based e-commerce product categorization. In *eCOM@SIGIR*.

Yu Zhang, Frank F Xu, Sha Li, Yu Meng, Xuan Wang, Qi Li, and Jiawei Han. 2019. Higitclass: Keyword-driven hierarchical classification of github repositories. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 876–885. IEEE.

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 Computational Complexity

We compare the computational complexity of PAAM-HiA-T5 with that of the best performing BERT+HiMatch and BERT+HiAGM. From the perspective of space complexity, PAAM-HiA-T5 contains roughly 220M parameters. BERT+HiMatch and BERT+HiAGM respectively contain about 153M and 143M parameters. The model sizes of these three models are in the same order of magnitude. Regarding time complexity, the time cost of PAAM-HiA-T5 is only about 0.82 times that of BERT+HiMatch and about 0.91 times that of BERT+HiAGM during training with batch size 10. In the inference stage, all three models run on the same GeForce RTX 2080 Ti GPU with their respectively highest GPU usage. Under this condition, PAAM-HiA-T5's total time cost for predicting

all test data is about 3 times that of BERT+HiMatch and BERT+HiAGM. PAAM-HiA-T5 establishes new SOTA results on HTC task, so we think that the enlargement in computational complexity due to the generative model properties is acceptable.

B Performance Comparison on Label Granularity with SOTA Methods

To further clarify the superiority of PAAM-HiA-T5, we perform the level-based performance analysis between our approach and other best performing SOTA methods on RCV1-V2. The level-based Micro-F1 scores and Macro-F1 scores are shown in Table 8. There is a dip in Micro-F1 score at second level for all models because there are lots of confusing labels with close concepts at second level. The relatively low Macro-F1 scores at the second and third levels are due to the presence of long-tailed labels. Figure 8 shows that our model maintains advanced performance on all levels, especially on lower levels. This reflects that our model has a huge advantage in dealing with lower-level long-tailed labels with sparse data.

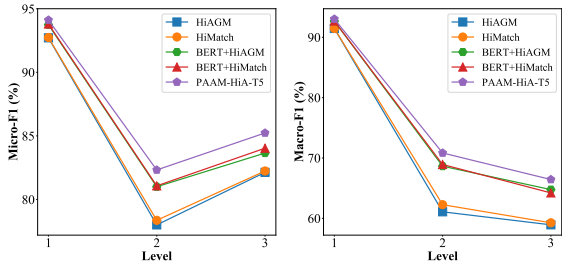

Figure 8: Performance analysis on label granularity based on different levels.

C Exploring the Impact of Hierarchy-Aware Module on the Pre-trained Base Model

We find that BERT+HiAGM, BERT+HiMatch and PAAM-HiA-T5 are most competitive methods according to previous experiments. On the one hand, the pre-trained models, including BERT and T5, can be viewed as the base models. On the other hand, different mechanisms and strategies, including HiAGM, HiMatch and PAAM-HiA, are utilized to exploit hierarchical structure information based on the pre-trained base models, and they can be regarded as different hierarchy-aware modules. We want to study the improvement of the pre-trained base models brought by different hierarchy-aware modules in HTC task.

Table 7 shows that the base models' performance of BERT+HiAGM, BERT+HiMatch and PAAM-HiA-T5 is close, both on Micro-F1 and Macro-F1. Therefore, it is fair to discuss the improvement brought by different hierarchy-aware modules to the pre-trained base model, and the performance changes are illustrated in Figure 9. For BERT+HiAGM, not only did the HiAGM not improve BERT's Micro-F1 score, it actually lowered the Micro-F1 score. The reason may be that HiAGM introduces noise in the process of encoding the overall hierarchy information. This degrades the performance of BERT+HiAGM on frequent labels. For BERT+HiMatch, HiMatch brings a relatively large improvement on Macro-F1, but a slight improvement on Micro-F1. This demonstrates that HiMatch has limited improvement for BERT on predicting middle-level and upper-level labels. But for PAAM-HiA-T5, PAAM-HiA module greatly boosts both Micro-F1 and Macro-F1 and establishes new SOTA results.

In conclusion, starting from base models with close performance, the improvement brought by the PAAM-HIA module to T5 significantly exceeds that brought by the HiMatch and HiAGM to BERT. Moreover, thanks to the PAAM-HiA module, our model outperforms all SOTA methods. All of the above fully illustrate that our mechanism and strategy (PAAM-HiA module), not just the powerful pre-trained base model, are important reasons for the strong power of our model in HTC task.

Ablation	BERT+HiAGM		BERT+HiMatch		PAAM-HiA-T5	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
BASE MODEL	86.26	67.35	86.26	67.35	86.14	67.39
+ Hierarchy-Aware Module	86.12	68.08	86.33	68.66	**87.22**	**70.02**

Table 7: Ablation study of hierarchy-aware modules on pre-trained base models. Specifically, "BASE MODEL" is either BERT or T5. "+ Hierarchy-Aware Module" denotes adding a hierarchy-aware module to the corresponding base model to obtain the final models.

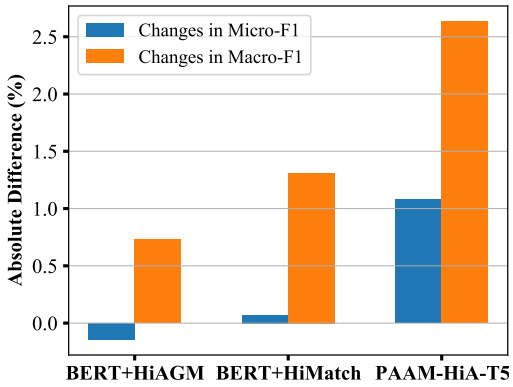

Figure 9: Model performance changes brought about by the hierarchy-aware modules. Specifically, the figure above shows the absolute difference between the performance of BERT+HiAGM, BERT+HiMatch and PAAM-HiA-T5 and that of BERT, BERT, and T5.