Hierarchy-aware Biased Bound Margin Loss Function for Hierarchical Text Classification

Anonymous ACL submission

Abstract

Hierarchical text classification (HTC) is a challenging problem with two key issues: utilizing structural information and mitigating label imbalance. Recently, the unit-based approach generating unit-based feature representations has outperformed the global approach focusing on a global feature representation. Nevertheless, unit-based models using BCE and ZLPR losses still face static thresholding and label imbalance challenges. Those challenges become more critical in large-scale hierarchies. This paper introduces a novel hierarchy-aware loss function for unit-based HTC models: Hierarchy-aware Biased Bound Margin (HBM) loss. HBM integrates learnable bounds, biases, and a margin to address static thresholding and mitigate label imbalance adaptively. Experimental results on benchmark datasets demonstrate the superior performance of HBM compared to competitive HTC models.

1 Introduction

Hierarchical Text Classification (HTC) aims to classify text into a predefined label hierarchy. HTC currently faces two fundamental challenges: utilizing structural information and mitigating label imbalance. As shown in Figure 1, recent research can be categorized into global and unit-based approaches based on exploiting feature representations combined with text and structural information. The global approach, HiAGM (Zhou et al., 2020), HiMatch (Chen et al., 2021), HGCLR (Wang et al., 2022a), K-HTC (Liu et al., 2023), HiTIN (Zhu et al., 2023), and HJCL (Yu et al., 2023), generates a holistic feature representation of text that encompasses an entire hierarchy and use it to compute label scores comprehensively. In contrast, the unit-based approach, HPT (Wang et al., 2022b) and HiDEC (Im et al., 2023), generates feature representations of text at the unit level, where a unit refers to a subset of a hierarchy partitioned by specific strategies, and classification is performed on labels within these units. Recently, the unit-based approach has achieved significant improvements over the global approach.

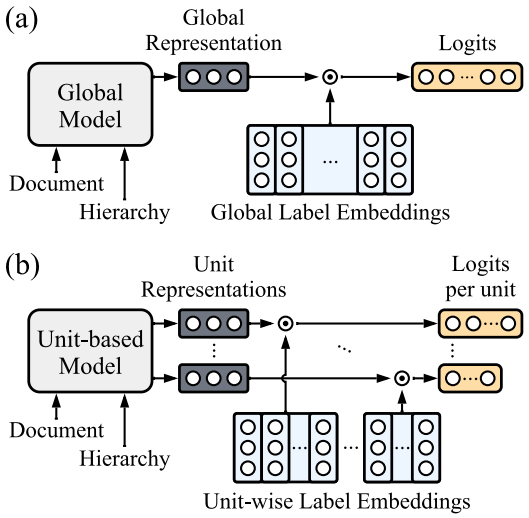

Figure 1: Classification processes of (a) Global and (b) Unit-based HTC models.

However, there are two significant limitations in existing research: static thresholding and label imbalance. Static thresholding is problematic because most HTC models utilizing binary cross entropy (BCE) loss predict positive labels using a fixed threshold, typically set at 0.5, when the output probability exceeds this threshold. Determining optimal thresholds for target labels is computationally intensive, particularly when considering various units. Label imbalance can lead to overtraining on frequent labels and undertraining on infrequent ones and degrade performance due to the dominance of numerous high-confidence labels in a loss. Previous HTC methods address this issue by leveraging hierarchical structures through the exploitation of auxiliary loss functions (Chen et al., 2021; Wang et al., 2022a; Liu et al., 2023; Yu et al., 2023) or per-unit classification (Kowsari et al., 2017; Banerjee et al., 2019; Shimura et al., 2018; Wang et al., 2022b; Im et al., 2023). Re-

cently, HPT and HJCL presented Zero-bounded Log-sum-exp Pairwise Rank-based (ZLPR) loss (Su et al., 2022), but the static thresholding problem still remains.

To tackle these limitations, this paper introduces a novel hierarchy-aware loss function for unit-based HTC models: Hierarchy-aware Biased Bound Margin (HBM) loss. Our key innovations in HBM are summarized as follows. First, we introduce learnable bounds for all units within a hierarchy to address the static thresholding problem. These bounds are optimized for various units during training and serve as dynamic unit thresholds during inference. Second, we introduce biases and a margin to mitigate the label imbalance. The biases promote low-confidence labels by adjusting the bounds of positive and negative label sets. These adjustments are dynamically determined based on the label logits and amplify the importance of undertrained labels. In contrast, the margin aims to diminish the domination of overtrained labels in a loss by excluding high-confidence labels.

Through a series of experiments, we demonstrate the effectiveness of our loss function applied to recent unit-based HTC models, HPT and HiDEC, using three benchmark datasets: RCV1-v2 (Lewis et al., 2004), NYT (Sandhaus, 2008), and EURLEX57K (Chalkidis et al., 2019). Notably, our loss function outperforms competitive HTC models on all three benchmark datasets. We comprehensively analyze how the bounds address the issue of static thresholding, and how the biases and margin address the problem of label imbalance in HTC.

Our contributions are summarized as follows:

- We propose a novel hierarchy-aware loss function, HBM, for unit-based HTC models to address static threshold and label imbalance by introducing bounds, biases, and a margin. The bounds are optimized during training and used as dynamic unit thresholds during inference. The biases and margin mitigate label imbalance by promoting low-confidence labels and excluding high-confidence labels from a loss, respectively.

- We demonstrate the effectiveness of our loss function applied to recent unit-based HTC models by comparing competitive HTC models on three benchmark datasets. Our results confirm the superiority and behaviors of our loss function, supported by in-depth analysis.

2 Related work

Recent HTC research based on deep learning can be categorized into global and unit-based approaches, each with its unique way of creating feature representations that incorporate both text and hierarchy structure.

The unit-based approach generates feature representations at the unit level by partitioning the entire hierarchy into units using specific strategies. Each unit corresponds to a subset of labels within a hierarchy. Various models employ diverse unit construction strategies, including "for-each-class" (Banerjee et al., 2019), "for-each-parent" (Kowsari et al., 2017; Im et al., 2023), "for-each-level" (Shimura et al., 2018; Wang et al., 2022b), and "for-each-sub-hierarchy" (Peng et al., 2018). HDLTex (Kowsari et al., 2017) introduces HTC models using DNN, CNN, and RNN architectures. HTrans (Banerjee et al., 2019) enhances HDLTex by employing transfer learning to preserve path information. HR-DGCNN (Peng et al., 2018) utilizes recursive hierarchical segmentation to divide a hierarchy into sub-hierarchies and construct local unit models. However, the unit-based approach often suffers from a lack of hierarchical information.

In contrast, the global approach generates a holistic feature representation encompassing the entire label hierarchy. HiAGM (Zhou et al., 2020) merges text and structural representations through text propagation, while HGCLR (Wang et al., 2022a) propagates structural representation through a text encoder and employs contrastive learning. HiMatch (Chen et al., 2021) applies a hierarchy-aware matching loss to HiAGM and adjusts feature representations based on hierarchy information. K-HTC (Liu et al., 2023) tries to incorporate a knowledge graph into HTC using knowledge-aware hierarchical label attention and contrastive learning. HiTIN (Zhu et al., 2023) reduces the complexity of the existing global models by reconstructing a hierarchy to minimize structural entropy. HJCL (Yu et al., 2023) proposes a unified loss function integrating instance and label-wise contrastive learning losses, along with ZLPR loss (Su et al., 2022). The global models effectively leverage hierarchical information through structure encoders (Kipf and Welling, 2017; Ying et al., 2021), outperforming unit-based models. Despite their achievements, they face challenges of label imbalances and hierarchy-dependent model parameters.

To address these challenges, HPT (Wang et al.,

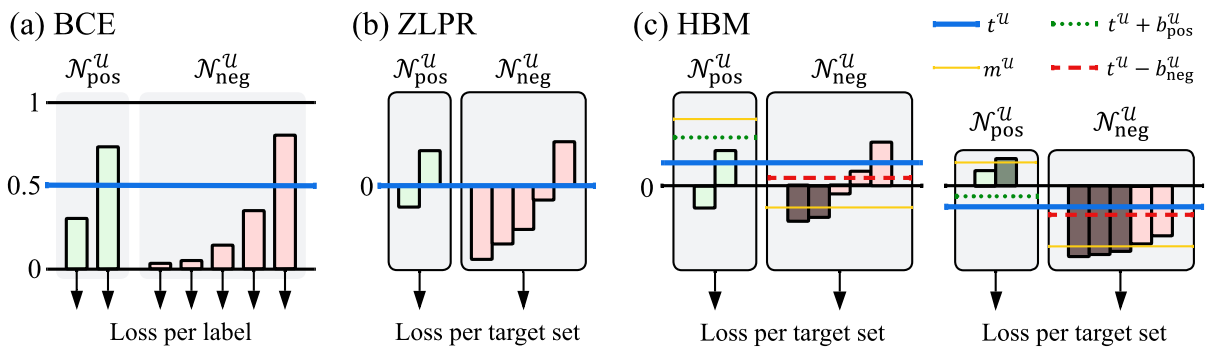

Figure 2: The illustration of classification losses, (a) BCE, (b) ZLPR, and (c) HBM. The blue line is a threshold during inference. In BCE, a loss is computed for each label and then averaged. In ZLPR and HBM, a loss is calculated separately for positive and negative target sets and combined. The green and red lines are positive and negative biased bounds, respectively, during training. In HBM, a bound is optimized for each unit and used as dynamic thresholds during inference. The yellow lines represent a margin. Labels exceeding the margin are excluded in computing a loss.

2022b) and HiDEC (Im et al., 2023) incorporate a structure encoder (Veličković et al., 2018) and attention mechanism (Vaswani et al., 2017) into their unit-based HTC models. HiDEC utilizes an encoder-decoder architecture to generate a sub-hierarchy sequence based on the target labels of each document using a parent-level unit construction strategy. By dividing a hierarchy based on levels, HPT integrates level-specific feature representations from a structure encoder into a text encoder and proceeds with unit-wise prediction. Furthermore, HPT incorporates ZLPR loss by introducing a zero-bound to MLCE loss (Li et al., 2017; Sun et al., 2020). With ZLPR loss, HPT transforms HTC into a multi-label masked language modeling task. However, these methods still encounter label imbalance in large-scale hierarchies and suffer from static thresholding.

3 Proposed Hierarchy-aware Loss Function

3.1 Preliminaries and Notations

Let a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a predefined hierarchy where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a set of all label nodes and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ is a set of edges indicating a relation between two nodes. $\mathcal{D} = \{(x_d, \mathcal{Y}_d)\}_{d=1}^{|D|}$ is a document dataset where x_d is d-th document and $\mathcal{Y}_d \subset \mathcal{V}$ is a set of target labels associated with x_d. Note $|\mathcal{Y}_d| \geq 1$ because a document x_d can have multi-labels. We partition $\mathcal{V}$ into a set of units $\mathcal{W} = \{\mathcal{U}_1, \ldots, \mathcal{U}_{|\mathcal{W}|}\}$ where $\mathcal{U}$ denotes a unit composed of a set of labels.

For a given document x_d, unit-based HTC models generate a unit representation $\boldsymbol{r}^{\mathcal{U}}$, then compute logits $\boldsymbol{l}^{\mathcal{U}}$ using the unit representation $\boldsymbol{r}^{\mathcal{U}}$ and label embeddings associated with the labels in a unit $\mathcal{U}$. These logits $\boldsymbol{l}^{\mathcal{U}}$ are used to make predictions on a unit $\mathcal{U}$. The target label set for each unit is defined as $\mathcal{Y}_d^{\mathcal{U}} = \{v_i | v_i \in (\mathcal{Y}_d \cap \mathcal{U})\}$.

To calculate a loss, we divide a unit $\mathcal{U}$ into positive and negative target sets, denoted as $\mathcal{N}^{\mathcal{U}}_{\text{pos}} = \{v_i | v_i \in \mathcal{Y}_d^{\mathcal{U}}\}$ and $\mathcal{N}^{\mathcal{U}}_{\text{neg}} = \{v_i | v_i \in \mathcal{U} \backslash \mathcal{Y}_d^{\mathcal{U}}\}$. If the target label does not exist within a specific unit, $\mathcal{N}^{\mathcal{U}}_{\text{pos}}$ can become an empty set. Based on $\mathcal{N}^{\mathcal{U}}_{\text{pos}}$ and $\mathcal{N}^{\mathcal{U}}_{\text{neg}}$ in Figure 2-(a), BCE loss is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{\sum_{\mathcal{U} \in \mathcal{W}} |\mathcal{U}|} \sum_{\mathcal{U} \in \mathcal{W}} \left[\sum_{p \in \mathcal{N}^{\mathcal{U}}_{\text{pos}}} \log \sigma(l_p^{\mathcal{U}}) + \sum_{n \in \mathcal{N}^{\mathcal{U}}_{\text{neg}}} \log(1 - \sigma(l_n^{\mathcal{U}})) \right], \tag{1}$$

where $l_p^{\mathcal{U}}$ and $l_n^{\mathcal{U}}$ are the logits for positive label p and negative label n, respectively. $\sigma(\cdot)$ is a sigmoid function.

BCE loss has a weakness in dealing with label imbalance. To this end, ZLPR (Zero-bounded Log-sum-exp Pairwise Rank-based) loss (Su et al., 2022) is presented in HPT (Wang et al., 2022b) :

$$\mathcal{L}_{\text{ZLPR}} = \frac{1}{|\mathcal{W}|} \sum_{\mathcal{U} \in \mathcal{W}} \left[\log(1 + \sum_{p \in \mathcal{N}^{\mathcal{U}}_{\text{pos}}} e^{-l_p^{\mathcal{U}}}) + \log(1 + \sum_{n \in \mathcal{N}^{\mathcal{U}}_{\text{neg}}} e^{l_n^{\mathcal{U}}}) \right]. \tag{2}$$

As depicted in Figure 2-(b), ZLPR loss attempts to mitigate label imbalance through the log-sum-exp operation to reduce the dominance of over-trained labels. However, it does not address static

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{W}\rvert$	Level	Average of					Train	Dev	Test
				$\lvert\mathcal{Y}_d\rvert$	$\lvert\mathcal{W}_d\rvert$	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{N}^{\mathcal{U}}_{\text{pos}}\rvert$	$\lvert\mathcal{N}^{\mathcal{U}}_{\text{neg}}\rvert$			
RCV1-v2	103	4/22	4	3.24	4/2.98	25.75/5.63	0.80/1.77	24.95/3.86	20,833	2,316	781,265
NYT	166	8/52	8	7.60	8/6.94	20.75/4.17	0.95/1.79	19.80/2.38	23,345	5,834	7,292
EURLEX57K	4,271	6/1,168	6	5.00	6/9.16	752.17/5.15	0.85/1.06	751.32/4.09	45,000	6,000	6,000

Table 1: Data statistics. Level and $|\mathcal{V}|$ are the maximum level and number of labels in a hierarchy, while $|\mathcal{W}|$ is the number of units. $|\mathcal{Y}_d|$ and $|\mathcal{W}_d|$ are the average number of target labels and units for a document, while $|\mathcal{U}|$ is the average number of labels in a unit. $|\mathcal{N}^{\mathcal{U}}_{\text{pos}}|$ and $|\mathcal{N}^{\mathcal{U}}_{\text{neg}}|$ are the average number of positive and negative labels for units, respectively. Note that values partitioned by '/' indicate HPT and HiDEC in order.

thresholding because the bounds for all units remain fixed at 0.

3.2 Hierarchy-aware Biased Bound Loss

We propose a Hierarchy-aware Biased Bound Margin (HBM) loss to simultaneously address the issues of static thresholding and label imbalance within a unit $\mathcal{U}$. HBM is defined as:

$$\mathcal{L}_{\text{HBM}} = \frac{1}{|\mathcal{W}|}\sum_{\mathcal{U}\in\mathcal{W}}[\log(1+\sum_{p\in\mathcal{N}'^{\mathcal{U}}_{\text{pos}}} e^{-l^{\mathcal{U}}_p+(t^{\mathcal{U}}+b^{\mathcal{U}}_{\text{pos}})}) + \log(1+\sum_{n\in\mathcal{N}'^{\mathcal{U}}_{\text{neg}}} e^{l^{\mathcal{U}}_n-(t^{\mathcal{U}}-b^{\mathcal{U}}_{\text{neg}})})], \tag{3}$$

where $t^{\mathcal{U}} \in \mathbb{R}$ is a learnable bound for a unit $\mathcal{U}$. $b^{\mathcal{U}}_{\text{pos}}$ and $b^{\mathcal{U}}_{\text{neg}}$ are positive and negative biases for a unit $\mathcal{U}$, respectively. $\mathcal{N}'^{\mathcal{U}}_{\text{pos}}$ and $\mathcal{N}'^{\mathcal{U}}_{\text{neg}}$ are positive and negative target sets, respectively, after excluding labels with a margin.

The bound $t^{\mathcal{U}}$ is computed using a unit representation $\boldsymbol{r}^{\mathcal{U}}$, allowing us to predict distinct bounds for each unit by leveraging text and hierarchy information. During training, we enforce that positive labels have higher logits than $t^{\mathcal{U}}$, whereas negative labels have vice versa. Subsequently, the optimized bound $t^{\mathcal{U}}$ is utilized as a dynamic threshold by $\hat{\mathcal{Y}}^{\mathcal{U}}_d = \{v_i | l^{\mathcal{U}}_{v_i} > t^{\mathcal{U}}, v_i \in \mathcal{U}\}$ during inference.

The biases $b^{\mathcal{U}}_{\text{pos}}$ and $b^{\mathcal{U}}_{\text{neg}}$ can be computed using any function $g : \mathcal{N} \to \mathbb{R}^+$ designed to promote training on the low-confidence labels in $\mathcal{N}^{\mathcal{U}}_{\text{pos}}$ and $\mathcal{N}^{\mathcal{U}}_{\text{neg}}$. We employ the standard deviation, $g = \alpha \cdot \text{std}(\{l^{\mathcal{U}}_v | v \in \mathcal{N}\})$ where α is a hyperparameter. Like Figure 2-(c), a high standard deviation of logits indicates insufficient model training on the labels within $\mathcal{N}$, leading to the assignment of higher biases. The bias adjusts the bound applied to positive labels higher and negative labels lower. These adjustments provide an opportunity to better train on low-confidence labels, influenced by the log-sum-exp function. Practically, biases $b^{\mathcal{U}}_{\text{pos}}$ and $b^{\mathcal{U}}_{\text{neg}}$ are computed with detached gradients on the target sets during training.

The margin m is applied to all labels before calculating the loss. A logit is transformed into a probability score $s_{v_i} = \sigma(2(l^{\mathcal{U}}_{v_i} - t^{\mathcal{U}}))$ according to Su et al. (2022). We redefine $\mathcal{N}'^{\mathcal{U}}_{\text{pos}} = \{v_i | s_{v_i} < 1-m, v_i \in \mathcal{N}^{\mathcal{U}}_{\text{pos}}\}$ and $\mathcal{N}'^{\mathcal{U}}_{\text{neg}} = \{v_i | s_{v_i} > m, v_i \in \mathcal{N}^{\mathcal{U}}_{\text{neg}}\}$ to retain labels participating in training with the margin. As shown in Figure 2-(c), labels unsatisfying with the margin are regarded as high-confidence labels dominating a loss and are removed. Consequently, it can mitigate the label imbalance.

3.3 Implementations on Unit-based Model

To validate the effectiveness of HBM loss, we have applied it to two recent unit-based HTC models, HPT (Wang et al., 2022b) and HiDEC (Im et al., 2023). These models employ distinct strategies for partitioning a hierarchy into a set of units. In HPT, the same units are utilized during both training and inference. In contrast, HiDEC exhibits variability in its units. This difference stems from the fact that in HiDEC, for a document x_d, units are constructed using the target label set $\mathcal{Y}_d$ during training, whereas during inference, units are formed through sub-hierarchy expansion starting from the root. Specifically, in HPT, each unit encompasses all labels at the same hierarchy level. We denote a unit and a target label set for the k-th level as $\mathcal{U}_k = \{v_i | \text{level}(v_i) = k, v_i \in \mathcal{V}\}$ and $\mathcal{Y}^{\mathcal{U}_k}_d = \{v_i | v_i \in \mathcal{Y}_d \cap \mathcal{U}_k\}$, respectively. In HiDEC, for a given document x_d, a sub-hierarchy label set $\mathcal{V}_d = \mathcal{Y}_d \cup \{v_i | v_i \in \text{ancestor}(v_j), v_j \in \mathcal{Y}_d\}$ and a sub-hierarchy sequence $\text{H}^d = [v_i | v_i \in \mathcal{V}_d \backslash \text{leaf}(\mathcal{G})]$ are created sequentially. Based on H^d, the k-th parent unit is defined as $\mathcal{U}_k = \{v_i | v_i \in \text{child}(\text{H}^d_k)\} \cup \{v_{\text{end}}\}$, where v_{end} is a special node used to terminate sub-

Model	Approach	Loss	RCV1-v2		NYT		EURLEX57K	
			Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
BERT†‡	Global	BCE	85.65†	67.02†	78.24†	65.62†	73.20‡	-
HiAGM (Zhou et al., 2020)	Global	BCE*	85.58	67.35	78.64	66.76	-	-
HiMatch (Chen et al., 2021)	Global	BCE*	86.33	68.66	-	-	-	-
HGCLR (Wang et al., 2022a)	Global	BCE*	86.49	68.31	78.86	67.96	-	-
HiTIN (Zhu et al., 2023)	Global	BCE*	86.71	69.95	79.65	69.31	-	-
HJCL (Yu et al., 2023)	Global	ZLPR*	87.04	**70.49**	**80.52**	70.02	-	-
HPT (Wang et al., 2022b)	Unit	ZLPR*	87.26	69.53	80.42	**70.42**	-	-
HiDEC (Im et al., 2023)	Unit	BCE	**87.96**	69.97	79.99	69.64	**75.29**	-
Our Implementations								
HPT	Unit	BCE*	$87.65_{\pm 0.11}$	$69.87_{\pm 0.40}$	$79.49_{\pm 0.22}$	$68.66_{\pm 0.30}$	$71.57_{\pm 0.58}$	$25.34_{\pm 0.59}$
		ZLPR*	$87.82_{\pm 0.14}$	$70.23_{\pm 0.31}$	$80.04_{\pm 0.23}$	$69.69_{\pm 0.49}$	$75.54_{\pm 0.20}$	$28.46_{\pm 0.26}$
		HBM*	$\mathbf{87.82}_{\pm 0.06}$	$\mathbf{70.55}_{\pm 0.13}$	$\mathbf{80.42}_{\pm 0.12}$	$\mathbf{70.23}_{\pm 0.18}$	$\mathbf{75.78}_{\pm 0.15}$	$\mathbf{28.70}_{\pm 0.22}$
HiDEC	Unit	BCE	$87.70_{\pm 0.12}$	$70.82_{\pm 0.20}$	$80.13_{\pm 0.16}$	$69.80_{\pm 0.24}$	$75.14_{\pm 0.19}$	$27.91_{\pm 0.11}$
		ZLPR	$87.59_{\pm 0.18}$	$70.61_{\pm 0.36}$	$80.25_{\pm 0.21}$	$70.14_{\pm 0.23}$	$76.16_{\pm 0.16}$	$28.68_{\pm 0.15}$
		HBM	$\mathbf{87.81}_{\pm 0.09}$	$\mathbf{71.47}_{\pm 0.20}$	$\mathbf{80.52}_{\pm 0.18}$	$\mathbf{70.69}_{\pm 0.19}$	$\mathbf{76.48}_{\pm 0.12}$	$\mathbf{28.77}_{\pm 0.11}$

Table 2: Overall performance. The upper shows the official scores reported in the original papers, whereas the lower presents the scores from our implementations, with each score accompanied by its standard deviation. Values are derived by averaging results from ten runs with random weight initialization. * indicates that an auxiliary loss is used with the classification loss, while _ represents the baseline loss for each model. † and ‡ denotes Wang et al. (2022a) and Chalkidis et al. (2019), respectively.

hierarchy expansion. Then, a target label set is defined as $\mathcal{Y}_d^{\mathcal{U}_k} = \{v_i | v_i \in \mathcal{V}_d \cap \mathcal{U}_k\}$. For a label assignment, we re-define $\mathcal{Y}_d^{\mathcal{U}_k} = \mathcal{Y}_d^{\mathcal{U}_k} \cup \{v_{\text{end}}\}$ if $\text{H}_k^d \in \mathcal{Y}_d$. In both HPT and HiDEC, a simple feed-forward network (FFN) is employed to learn optimal bounds based on unit representations. Consequently, HPT and HiDEC using HBM loss require only a modest number of additional parameters compared to the original models.

4 Experiments

4.1 Experimental settings

Datasets and Evaluation Metrics We selected two small-scale datasets, RCV1-v2 (Lewis et al., 2004) and NYT (Sandhaus, 2008), and a large-scale dataset, EURLEX57K (Chalkidis et al., 2019), for our standard experiments. To ensure a fair comparison, we adhered to the same data configuration as previous research (Zhou et al., 2020; Chen et al., 2021; Wang et al., 2022b; Im et al., 2023) and used Micro-F1 and Macro-F1 as our evaluation metrics. Table 1 presents the data statistics for three datasets. RCV1-v2 offers limited training data, while EURLEX57K provides a large number of labels. It is particularly noteworthy to examine the statistics of units. HPT (Wang et al., 2022b) generates a considerably smaller number of units compared to HiDEC (Im et al., 2023). We can see label imbalance explicitly as both HPT and HiDEC produce a limited number of positive but substantial negative labels. As a hierarchy size increases, label imbalance becomes pronounced in HPT, while it remains stable in HiDEC. NYT has the lowest average number of negative target labels $|\mathcal{N}_{\text{neg}}^{\mathcal{U}}|$.

Implementation Details We implemented HBM, BCE, and ZLPR losses using the original codes[1] based on HPT and HiDEC. The same model architectures and hyperparameters of the model were utilized for all three datasets.

In HPT, `bert-base-uncased` (Devlin et al., 2019) and GAT (Veličković et al., 2018) were used as text and structure encoders, respectively. The batch size was set to 16. Adam (Kingma and Ba, 2015) optimizer was used with a learning rate of 3e-5. The early stop was applied when Macro-F1 for developments set after each epoch did not increase during 6 epochs. The other hyperparameters were not tuned.

In HiDEC, `bert-base-uncased` was used as a text encoder, while a 2-layer transformer decoder (Vaswani et al., 2017) was used as a hierarchy decoder. The label embeddings were initialized using a normal distribution with $\mu = 0$ and $\sigma = 768^{-0.5}$. The batch size was set to 64. AdamW (Loshchilov and Hutter, 2019) optimizer was used with the learning rate 5e-5. The learning rate was scheduled using a linear scheduler with a warmup rate of 0.1 over 100 epochs.

[1]Check out code repositories referred to in HPT and HiDEC papers.

Model	Bounds	Biases	Margin	RCV1-v2		NYT		EURLEX57K	
				Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
HPT	-	-	-	87.82	70.23	80.04	69.69	75.54	28.46
	◯	-	-	87.78	**70.56**	80.20	70.04	75.69	28.51
	-	◯	-	**87.91**	70.30	80.26	70.12	75.78	28.47
	-	-	◯	87.70	70.43	80.21	69.83	75.74	**28.71**
	◯	◯	◯	87.82	70.55	**80.42**	**70.23**	**75.78**	28.70
HiDEC	-	-	-	87.59	70.61	80.25	70.14	76.16	28.68
	◯	-	-	87.70	70.91	80.28	70.18	76.17	28.68
	-	◯	-	87.43	71.01	80.38	70.51	76.35	28.58
	-	-	◯	87.71	70.80	80.46	70.45	76.31	**28.84**
	◯	◯	◯	**87.81**	**71.47**	**80.52**	**70.69**	**76.48**	28.77

Table 3: Ablation results of HBM on three datasets. All performance represents the average of 10 runs with random weight initialization. The top-ranked scores in each metric are highlighted in red-bolded, while the second-ranked scores are underlined.

The bias scale factor α was set to 0.1 and 1.0 for HPT and HiDEC, respectively (See Appendix A.2). The margin m was set to 0.1 for RCV1-v2 and NYT whereas 0.01 for EURLEX57K. Note that the bias in the final layer of the FFN, employed for predicting learnable bounds, was removed.

Comparison Models For comparison, we selected recent HTC models that leverage pre-trained language models: HiAGM (Zhou et al., 2020), HiMatch (Chen et al., 2021), HGCLR (Wang et al., 2022a), HiTIN (Zhu et al., 2023), HJCL (Yu et al., 2023), HPT (Wang et al., 2022b), and HiDEC (Im et al., 2023).

HiAGM: HiAGM utilizes the prior probability of parent-child label dependency as adjacency of Graph Convolution Networks (GCN) (Kipf and Welling, 2017). A text representation from a text encoder such as TextRCNN (Lai et al., 2015) or BERT is propagated to GCN using text propagation.

HiMatch: HiMatch considers HTC as a semantic matching problem and conducts text and label semantic matching to HiAGM through a hierarchy-aware matching loss. In addition, the hierarchy-aware margin loss learns to adjust the distance based on the label's hierarchical relationship to reflect hierarchy in presentation.

HGCLR: HGCLR points out the limitations of the existing models that use separate text and structure encoders and proposes a contrastive learning method that can inject structural information into the text encoder.

HiTIN: To address the limitations of the existing global approach, HiTIN employs a strategy of reconstructing the hierarchy into a code tree to reduce structural complexity effectively. This code tree construction aims to minimize structural entropy, resulting in a simplified hierarchy that maximizes the retention of structural information from the original hierarchy.

HJCL: To improve upon the limitations of existing contrastive learning HTC methods, HJCL proposes a supervised contrastive approach integrating instance-wise and label-wise contrastive learning. It utilizes propagated label embeddings from PLM and GAT to enable contrastive learning and classification training from both the instance and label perspectives.

HPT: HPT is the first attempt to address HTC using prompt tuning. It transforms HTC into a hierarchy-aware multi-label MLM to incorporate the HTC and MLM. The hierarchy representation at different levels, represented through GAT, is used in conjunction with text as input to BERT. Classification is performed for labels corresponding to units at each level.

HiDEC: To address the issue of excessive parameters in the existing models, HiDEC employs a sub-hierarchy composed of labels related to documents rather than the entire hierarchy. HiDEC transforms HTC into a sequence generation problem and conducts training to generate sub-hierarchy sequences.

HiAGM, HiMatch, HGCLR, HiTIN, and HJCL are global models, whereas HPT and HiDEC are unit-based models. All models employ BERT as a text encoder. Except for HJCL and HPT, which utilize ZLPR loss, the other models use BCE loss.

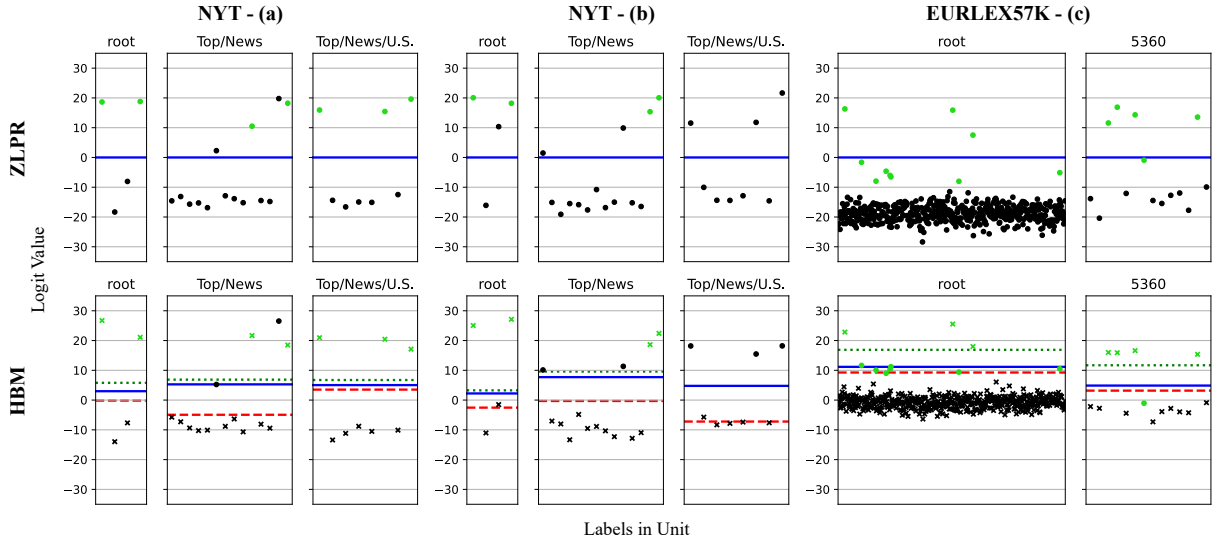

Figure 3: Illustration of sample logits of documents obtained from ZLPR and HBM during inference (a, b and c) in HiDEC. Each point on the graph represents a logit, with target labels in green and non-target labels in black, respectively. Blue lines denote the threshold used in each unit, while the green and red lines indicate positive and negative biased bounds, respectively, in HBM. The X marks denote the logits of labels excluded by the margin. In (a), HBM effectively reduces false predictions through dynamic thresholding. In (b), logits obtained with HBM are clearly distinguishable beyond the biased bounds. In (c), higher thresholds are observed when a unit comprises many few-shot labels.

4.2 Results

Table 2 presents the overall performance of three datasets. The scores and their variances were obtained from our implementations by averaging results from 10 runs with random weight initialization. HBM consistently achieved better performance over BCE and ZLPR on three datasets. It reveals that HBM is more effective on a large-scale HTC than a small-scale HTC, as the largest improvements were obtained from both HPT and HiDEC on EURLEX57K. It is worth to note that HBM is a single loss function designed to mitigate label imbalance compared to the existing research utilizing auxiliary loss functions.

4.3 Ablation studies

We conducted ablation studies to analyze the impact of bounds, biases, and a margin in HBM and summarized the results in Table 3. In Equations 2 and 3, HBM is equivalent to ZLPR when the bounds are set to 0, biases are removed, and no margin is applied. The biases are effective for all settings, particularly on RCV1-v2 and NYT. On EURLEX57K, the biases and margin improve Micro-F1 and Macro-F1, respectively. Compared to the biases and margin, the bounds alone seem insignificant. However, the combination of the three components complements each other and achieves improvements. Notably, HiDEC significantly benefits from HBM.

4.4 Analysis of Bounds, Biases, and Margin

Figure 3 illustrates the samples of thresholds, biased bounds, and logits obtained from ZLPR and HBM losses during inference in HiDEC. Each point on the graph represents a logit, with target labels in green and non-target labels in black, respectively. The X marks represent the logits of labels excluded by the margin from loss calculation. The logits are obtained from test documents in NYT (a and b) and EURLEX57K (c) using HiDEC's units. Blue lines denote the threshold used in each unit. In HBM, a threshold is determined by a bound predicted for each unit based on a specific document. The green and red lines indicate positive and negative biased bounds of units, respectively.

As in Figure 3, HBM employs dynamic thresholds for each document, whereas ZLPR applies a zero threshold to all units. We can see that the dynamic thresholds reduce false predictions for the specific units. However, the bounds optimized on a training set may be suboptimal. See Top/News on NYT and root on EURLEX57K in Figures 3-(b) and 3-(c). If Top/New/U.S. on NYT in Figure 3-(b) is used for training, only the three negative labels above the bound participate in the loss calculation

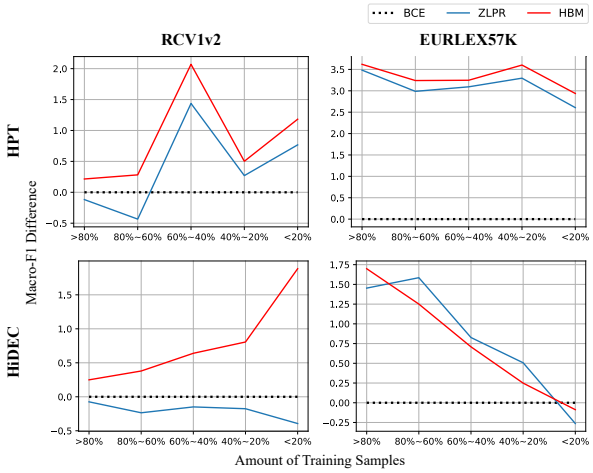

Figure 4: The Macro-F1 score differences on the test set for label clusters in two datasets. Label clusters are constructed by sorting in reverse order of frequencies and dividing them into five equal parts. The graph illustrates the performance difference of each model compared to models using BCE loss for each label cluster.

and fall below the biased bound of negative labels, the red line. Similarly, in the root on EURLEX57K of Figure 3-(b), only positive labels near the bound contribute to a loss and go above the green line.

High thresholds are observed with many negative target labels because increasing bounds is relatively easier than decreasing logits of all negative labels during training. So, infrequent positive labels struggle to increase logits than bounds. This issue can be mitigated by reducing the number of negative target labels, as observed on NYT.

4.5 Analysis of Label Imbalance

To analyze label imbalance, labels were clustered based on their frequency in the training set, and model performance was compared on these cluster. Label clusters were formed by sorting labels in reverse order of frequencies and dividing them into five equal parts. Simply, >80% cluster denotes frequent labels, whereas <20% is a cluster of infrequent labels. Figure 4 shows the performance difference on label clusters sorted by frequency. The black lines are the baselines for models trained with BCE. The red and blue lines indicate the score differences of HBM and ZLPR with respect to BCE, respectively. Notably, HBM effectively mitigates the label imbalance in most clusters as all score differences are positive and greater than those of ZLPR. On RCV1-v2, HBM leads to larger gains over BCE in all clusters. Specifically, the low-frequency clusters have more benefits than the high-frequency clusters. On EURLEX57K, HBM is effective, particularly for high-frequency clusters, but there is a diminishing tendency from high- to low-frequency clusters. This phenomenon stems from the fact that EURLEX57K has a long-tail distribution, with 83% of labels occurring less than 50 times.

5 Conclusion

This paper introduces a Hierarchy-aware Biased Bound Margin (HBM) loss function, offering two key innovations to address the challenges of static thresholding and label imbalance in HTC. First, HBM introduces learnable bounds for all units within a hierarchy to address static thresholding. These bounds are optimized for various units during training and are used as dynamic thresholds during inference. Second, HBM introduces biases and a margin to mitigate label imbalance. The biases promote low-confidence label training, while the margin excludes high-confidence labels from the loss. Third, Extensive experiments on benchmark HTC datasets demonstrate the superiority of HBM loss based on unit-based HTC models by comparing competitive HTC models and comprehensive analysis. We plan to extend HBM to extremely large-scale hierarchies and improve imbalance relations among units.

Limitations

When applying HBM to existing unit-based HTC models, additional parameters are required for optimizing learnable bounds. In this paper, we simply employed a single Feedforward Neural Network (FFN) with one hidden layer without exploration for model architecture because the FFN is guaranteed as it is widely used in various tasks. Any structure that allows dynamic optimization of bounds could enable the application of HBM.

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.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on EU legislation.

In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy. Association for Computational Linguistics.

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.

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.

SangHun Im, GiBaeg Kim, Heung-Seon Oh, Seongung Jo, and Dong Hwan Kim. 2023. Hierarchical text classification as sub-hierarchy sequence generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12933–12941.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. International Conference on Learning Representations, ICLR.

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

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.

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*, AAAI'15, page 2267–2273. AAAI Press.

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

Yuncheng Li, Yale Song, and Jiebo Luo. 2017. Improving pairwise ranking for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ye Liu, Kai Zhang, Zhenya Huang, Kehang Wang, Yanghai Zhang, Qi Liu, and Enhong Chen. 2023. Enhancing hierarchical text classification through knowledge graph integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5797–5810, Toronto, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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.

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.

Jianlin Su, Mingren Zhu, Ahmed Murtadha, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2022. Zlpr: A novel loss for multi-label classification.

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 (CVPR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022a. 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.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022b. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pages 28877–28888. Curran Associates, Inc.

Simon Chi Lok Yu, Jie He, Victor Basulto, and Jeff Pan. 2023. Instances and labels: Hierarchy-aware joint supervised contrastive learning for hierarchical multi-label text classification. pages 8858–8875. Association for Computational Linguistics.

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.

He Zhu, Chong Zhang, Junjie Huang, Junran Wu, and Ke Xu. 2023. HiTIN: Hierarchy-aware tree isomorphism network for hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7809–7821, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Dataset Details

We provide a more detailed examination of the datasets as presented in Table 1, yielding several key observations:

- **Train-Test Mismatch:** In RCV1-v2, there is a notable disparity in the sizes of the training and test sets, leading to a train-test mismatch.

- **Label Hierarchy Disparities:** EURLEX57K has a label hierarchy of 42 times and 25 times larger than RCV1-v2 and NYT, respectively. This substantial discrepancy in size causes a significant imbalance between positive and negative labels. The average of $|\mathcal{N}^{\mathcal{U}}_{\text{pos}}|$ remains relatively stable, while the average of $|\mathcal{N}^{\mathcal{U}}_{\text{neg}}|$ increases significantly from 24.95 and 19.80 in RCV1-v2 and NYT to 751.32 in EURELX57K.

- **Unit Imbalance:** The disparity in the unit construction strategies between HPT and HiDEC leads to substantial variations in unit statistics. HiDEC divides the hierarchy into smaller units than HPT, resulting in a small number of labels for each unit (Average of $|\mathcal{U}|$) and significantly balances the ratio of positive and negative labels for each unit (Average of $|\mathcal{N}^{\mathcal{U}}_{\text{pos}}|$ and $|\mathcal{N}^{\mathcal{U}}_{\text{neg}}|$). However, HiDEC still suffers from label imbalance.

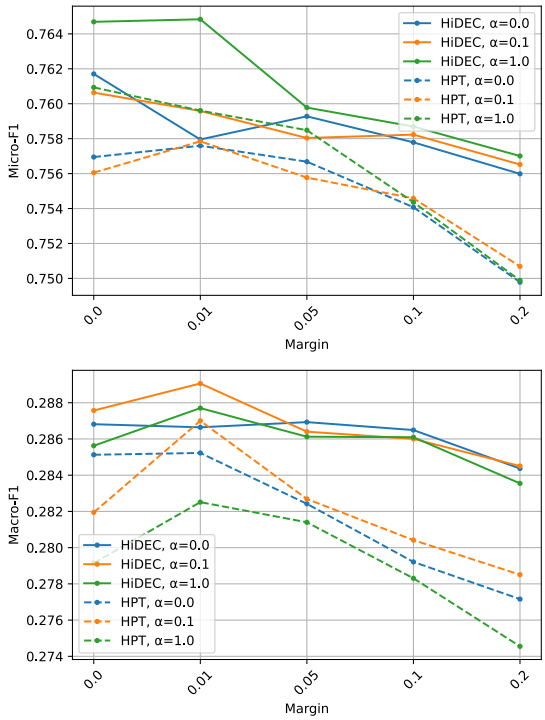

Figure 5: Hyperparameter exploration was conducted for EURLEX57K using HiDEC and HPT. All experiments averaged results from 5 runs with random weight initialization.

Additionally, EURLEX57K is categorized into three types based on label frequencies: "frequent" labels are those that appeared more than 50 times in the training data, "few-shot" labels are those that appeared less than 50 times, and "zero-shot" labels are those that have never appeared. This paper focuses on frequent and few-shot labels, as our baseline models, HPT and HiDEC, were not designed to handle zero-shot settings.

A.2 Exploration of hyperparameters

Figure 5 shows the hyperparameter exploration for HBM. For small-scale datasets, the bias scale factor α and margin m were heuristically set to 1.0 and 0.1 respectively. For the large-scale EURLEX57K dataset, experiments were conducted with varying combinations of α for bias and margin m. Based on these experiments, α and m of 1.0 and 0.01 respectively were chosen for HiDEC, which gave the highest Micro-F1 score. For HPT, α and m of 0.1 and 0.01 were selected, which provided the best Macro-F1 score.