HPT: Hierarchy-aware Prompt Tuning for Hierarchical Text Classification

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

Code： https://github.com/wzh9969/HPT.

(EMNLP-2022)

Reported by Zicong Dou

1. Introduction

2. Approach

3. Experiments

Introduction

Hierarchical Text Classification (HTC)

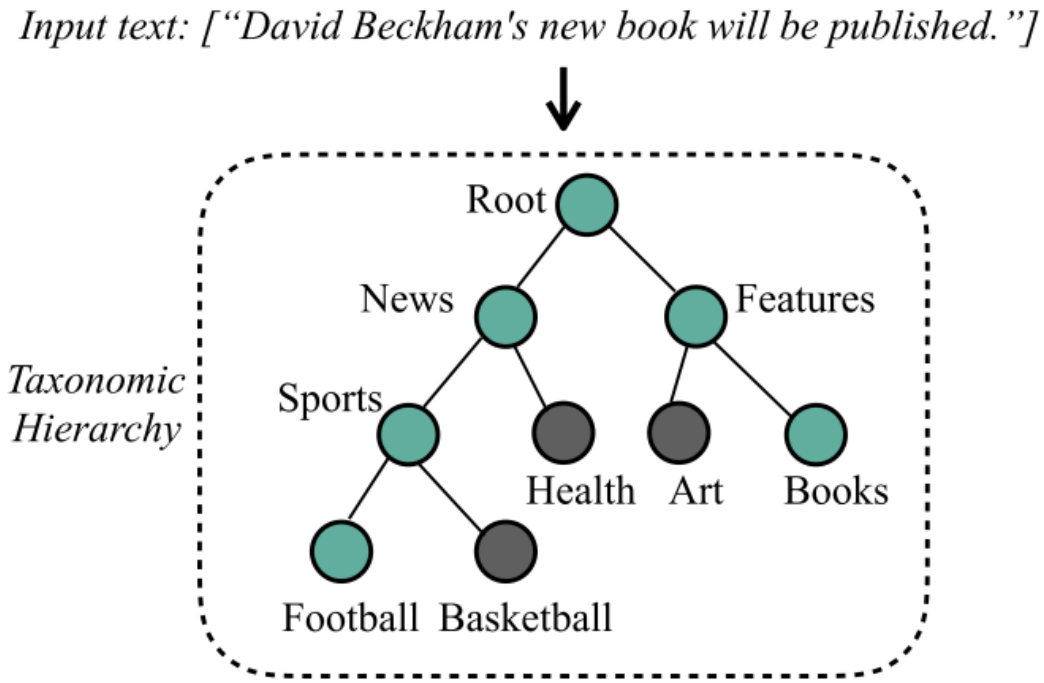


Figure 1: This short sample is tagged with *news*, *sports*, *football*, *features* and *books*. Note that HTC could be either a single-path or a multi-path problem.

Problem Definition

$$\mathcal{H} = (\mathcal{Y}, E) \quad Y \subseteq \mathcal{Y}.$$

Approach

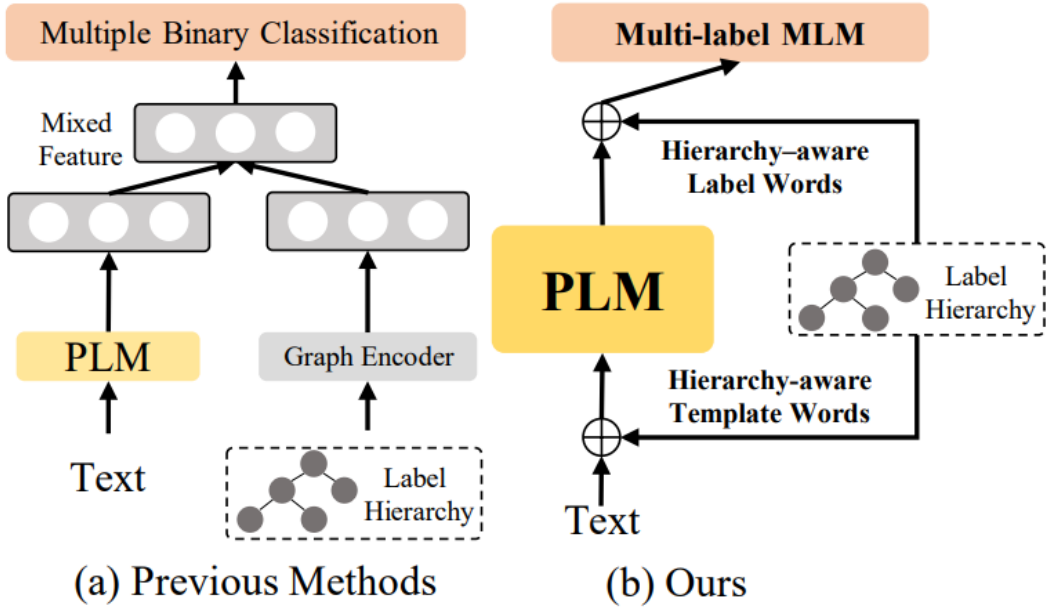


Figure 1: Comparison of previous methods and our HPT. (a) Previous models formulate HTC as a multiple binary classification problem, and utilize the PLM in a fine tuning paradigm . (b) HPT follows a prompt tuning paradigm that transforms HTC into a hierarchy-aware multi-label MLM problems.

Vanilla Fine Tuning for HTC

`[CLS]` **x** `[SEP]` $\mathbf{h}_{\texttt{[CLS]}}$

Prompt Tuning for HTC

x is `[MASK]`

Hard prompt	`[CLS]` **x** `[SEP]` The text is about `[MASK]` `[SEP]`
Soft prompt	`[CLS]` **x** `[SEP]` `[V1]` `[V2]` ... `[VN]` `[MASK]` `[SEP]`

(1)Hierarchy and flat gap.

(2)Multi-label and multi-class gap.

Approach

Step 1: prompt construction【Template】

Input: x = I love this movie.

Template: [x] Overall, it was a [z] movie.

Prompting: x' = I love this movie. Overall, it was a [z] movie.

Step 2: answer construction【Verbalizer】

☐ ☺ -> fantastic

☐ ☹ -> boring

label *answer*

Step 3: answer prediction【Prediction】

Input: x = I love this movie.

Template: [x] Overall, it was a [z] movie.

Answer: {fantastic:☺, boring:☹}

Prompting: x' = I love this movie. Overall, it was a [z] movie.

Predicting: x' = I love this movie. Overall, it was a fantastic movie.

Step 4: answer-label mapping【Mapping】

Input: x = I love this movie.

Template: [x] Overall, it was a [z] movie.

Answer: {fantastic:☺, boring:☹}

Prompting: x' = I love this movie. Overall, it was a [z] movie.

Predicting: x' = I love this movie. Overall, it was a fantastic movie.

Mapping: fantastic =>☺

Approach

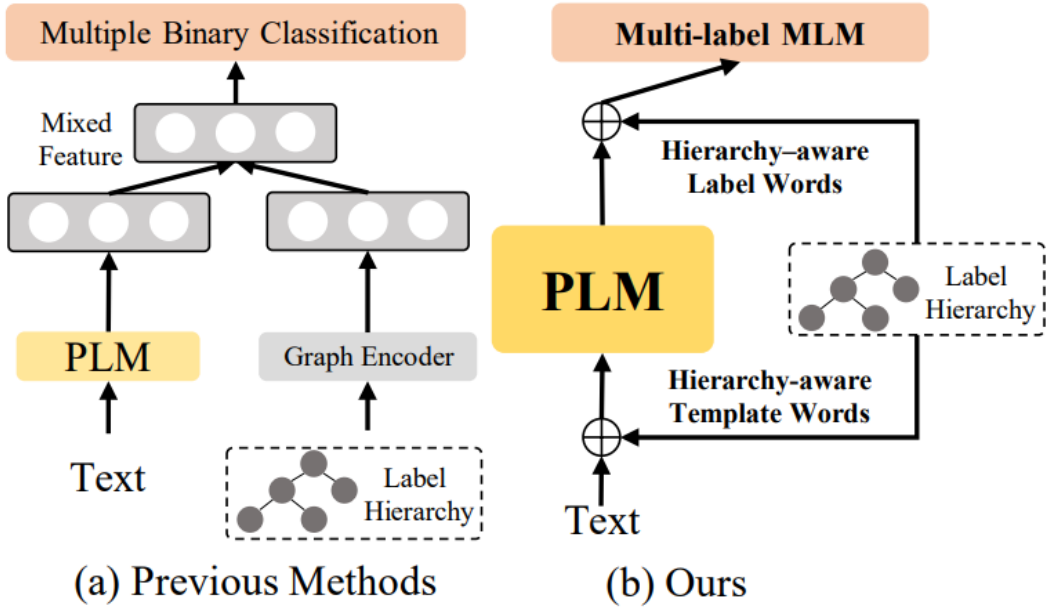


Figure 1: Comparison of previous methods and our HPT. (a) Previous models formulate HTC as a multiple binary classification problem, and utilize the PLM in a fine tuning paradigm . (b) HPT follows a prompt tuning paradigm that transforms HTC into a hierarchy-aware multi-label MLM problems.

Vanilla Fine Tuning for HTC

`[CLS]` **x** `[SEP]` $\mathbf{h}_{\texttt{[CLS]}}$

Prompt Tuning for HTC

x is `[MASK]`

Hard prompt	`[CLS]` **x** `[SEP]` The text is about `[MASK]` `[SEP]`
Soft prompt	`[CLS]` **x** `[SEP]` `[V1]` `[V2]` ... `[VN]` `[MASK]` `[SEP]`

(1)Hierarchy and flat gap.

(2)Multi-label and multi-class gap.

Approach

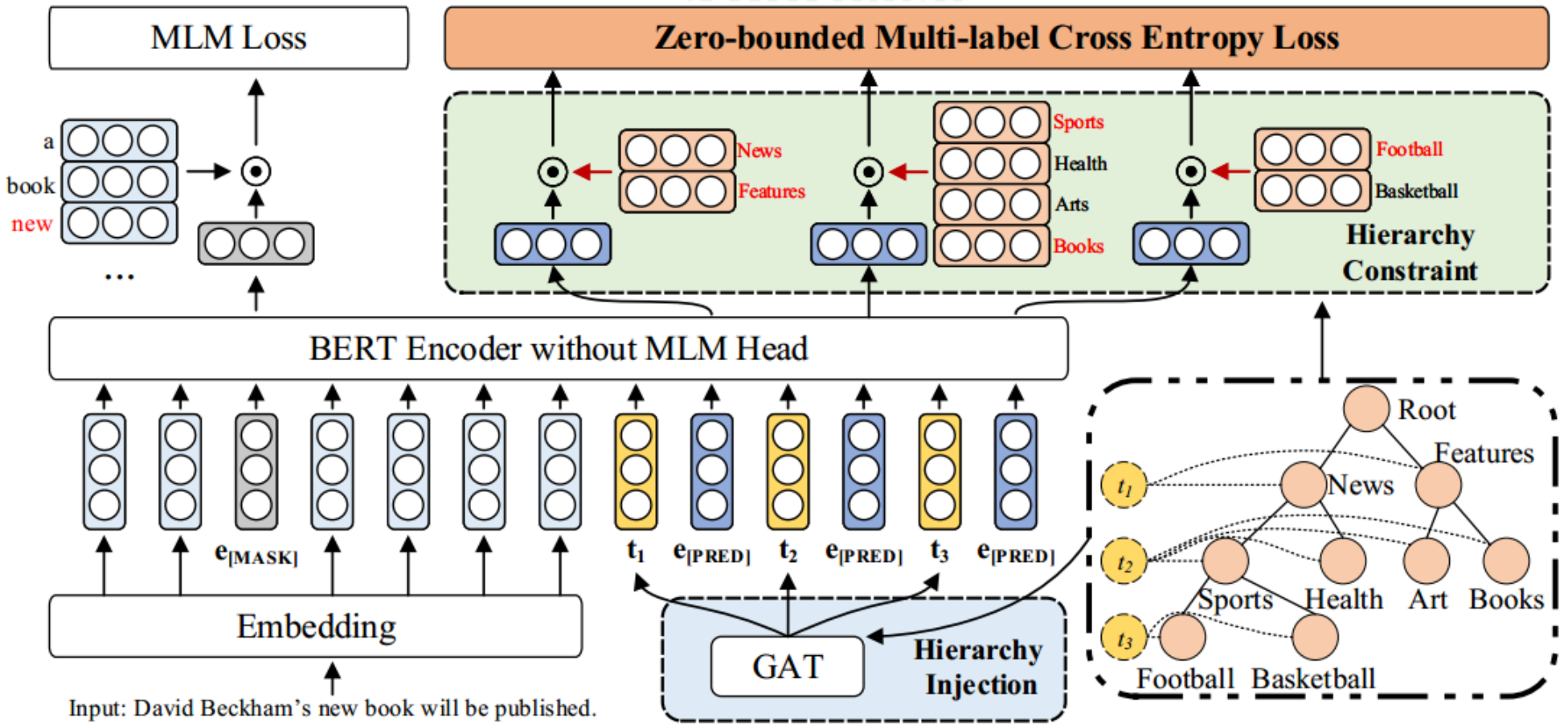
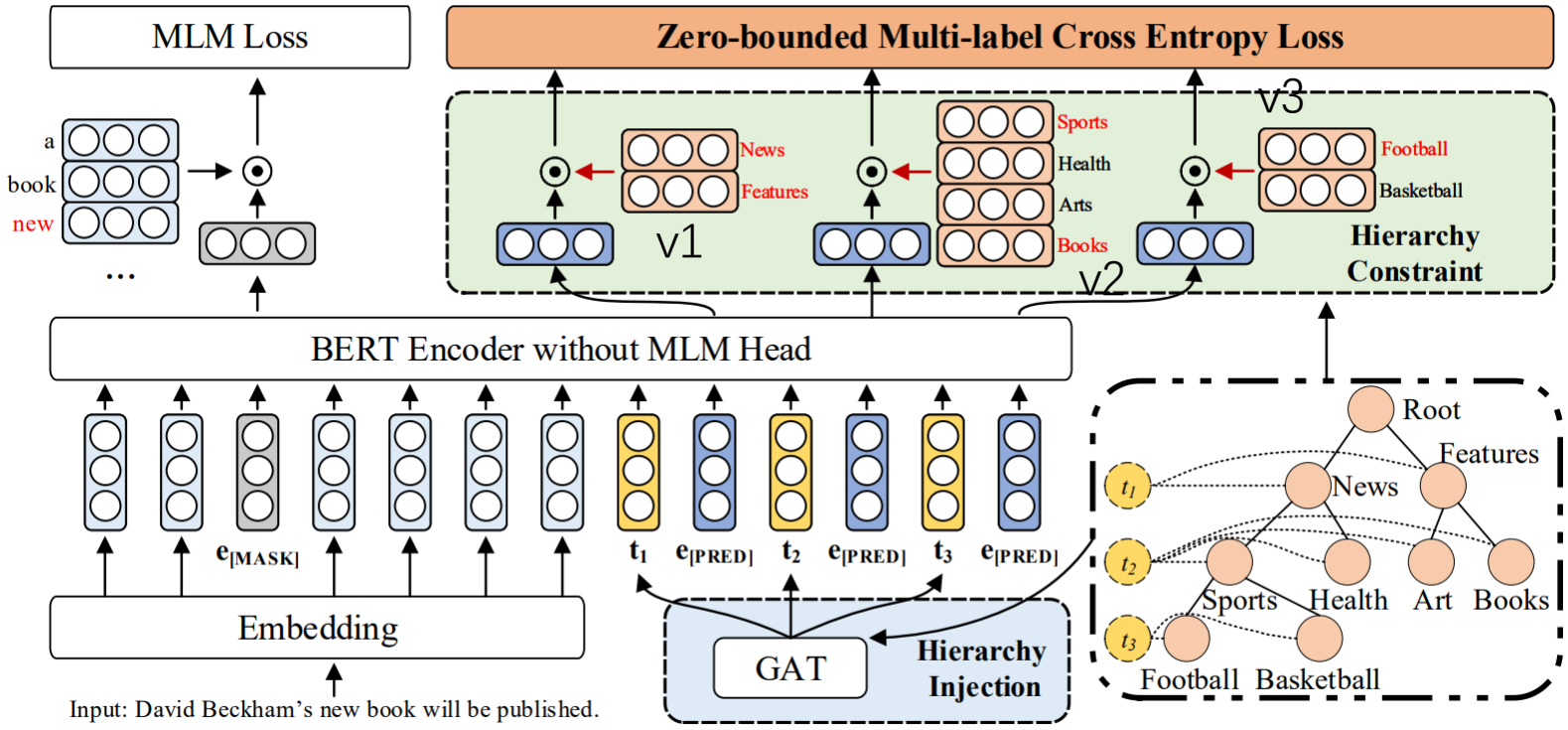


Figure 2: The architecture of HPT during training. HPT transforms HTC into a hierarchy-aware multi-label MLM problem that focuses on bridging *two* gaps between HTC and MLM. (1) To bridge the hierarchy and flat gap, HPT incorporates the label hierarchy knowledge into dynamic virtual template and label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

Approach



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

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

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

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

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

Hierarchy Constraint

$\mathcal{H} = (\mathcal{Y}, E)$ a depth of L input text $\mathbf{x}$

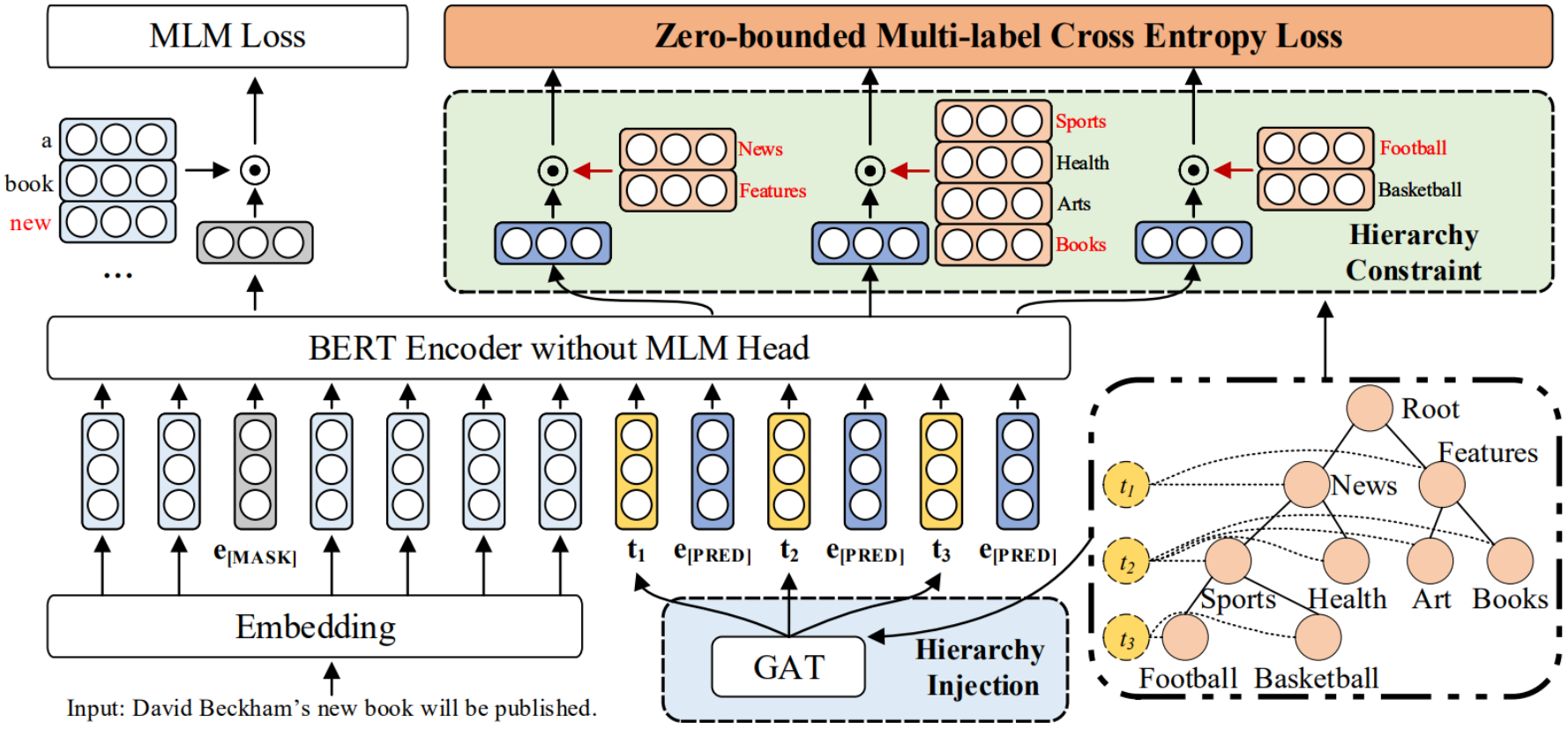
[CLS] **x** [SEP] [V1] [PRED]

[V2] [PRED] ... [VL] [PRED] [SEP]

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

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ is word embeddings and $\mathbf{e}_P$ is the embedding of [PRED], which is initialized by the [MASK] token of BERT. $\{\mathbf{t_i}\}_{i=1}^L$ are layer-wise template embeddings.

Approach



Hierarchy Injection

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

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

$t_1, \ldots, t_L$

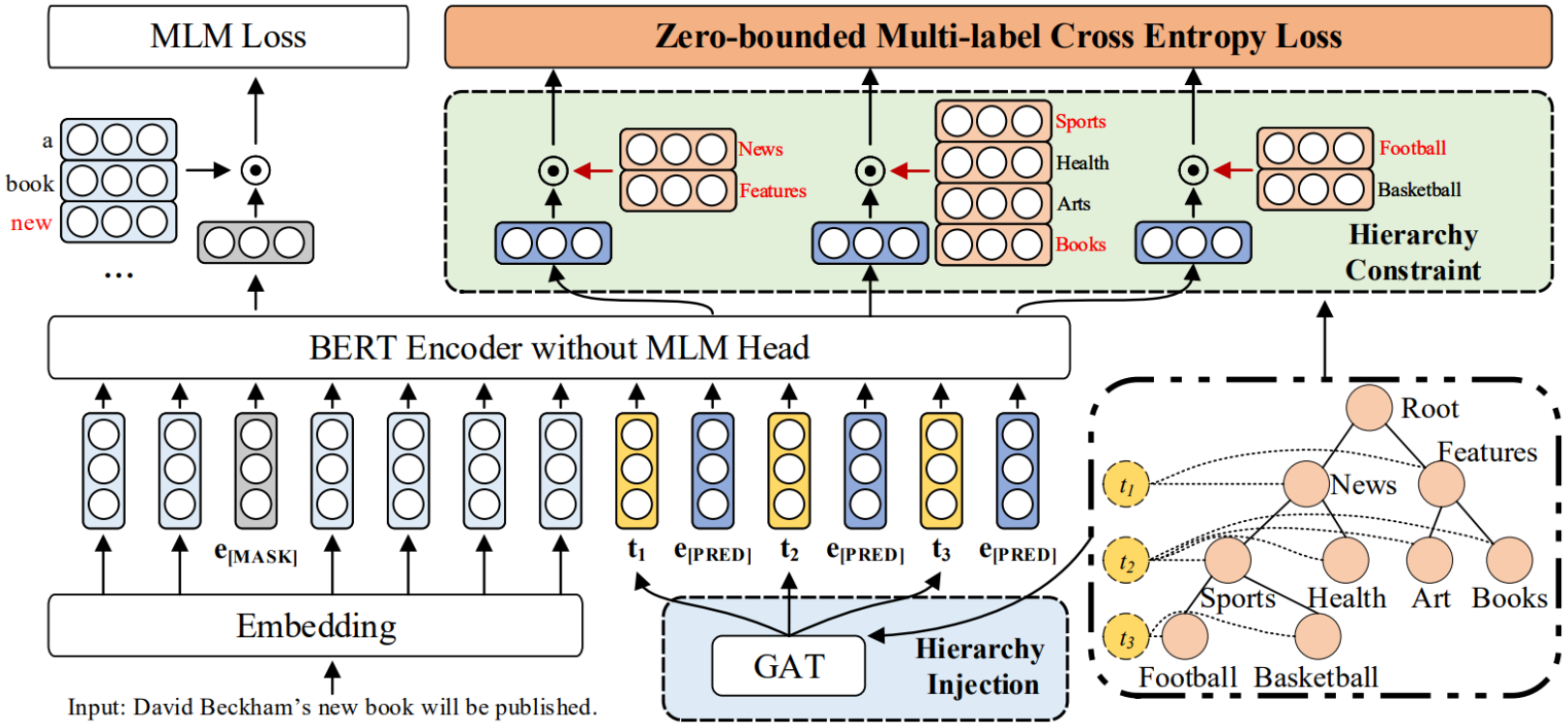
$\mathbf{v_i}$ $y_i \in \mathcal{Y}$

t_i $\mathbf{g}_{t_i}^K$

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

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

Approach



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

where s_{y_i} is the predicted sigmoid score of the label y_i for the input.

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

where y_t is the gold label for the input.

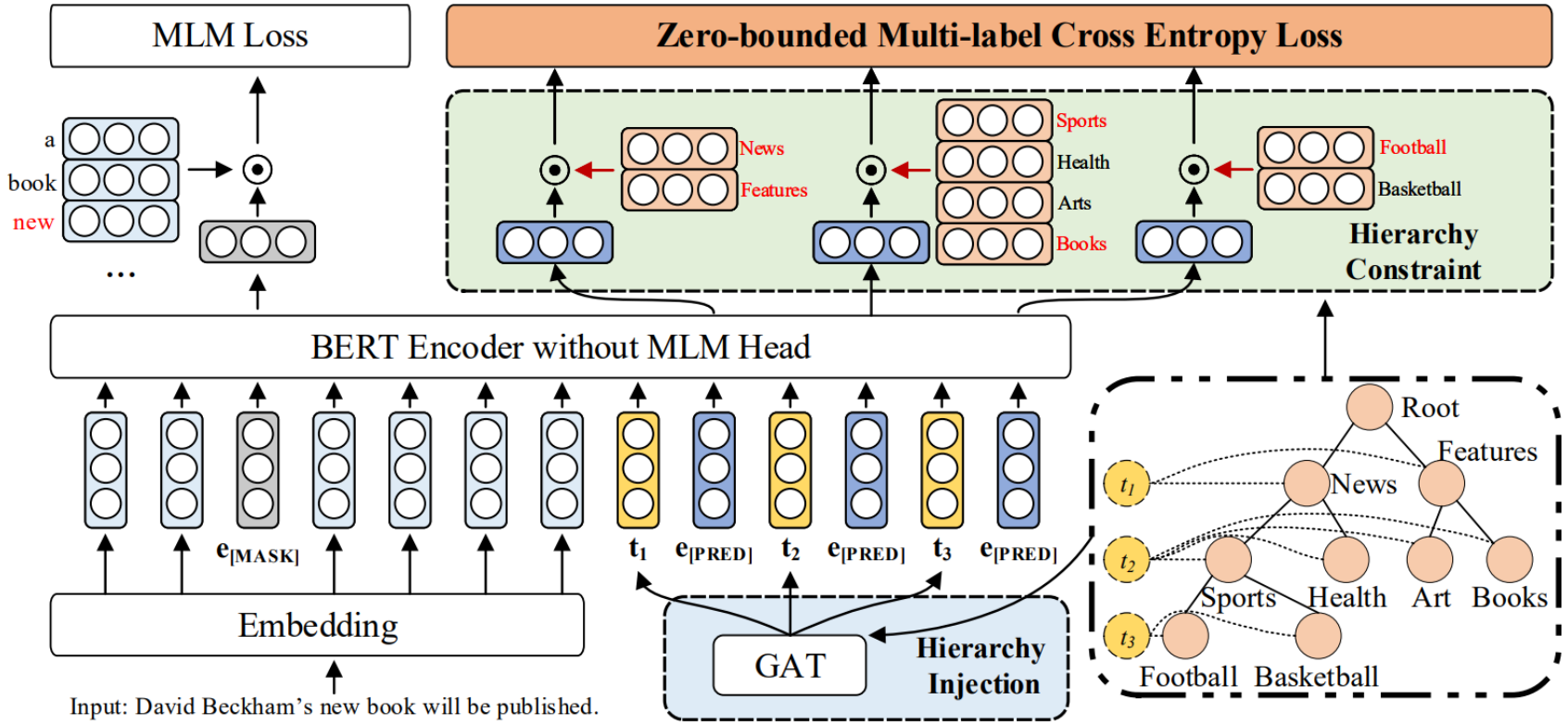
Zero-bounded Multi-label Cross Entropy Loss

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

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

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

Approach



$$\mathcal{L}^m_{ZMLCE} = \log(1 + \sum_{y_i \in \mathcal{N}^n_m} e^{s_{y_i}}) + \log(1 + \sum_{y_i \in \mathcal{N}^p_m} e^{-s_{y_i}}) \quad (10)$$

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

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

Experiments

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

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

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

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

Experiments

Model	WOS (Depth 2)		RCV1-V2 (Depth 4)		NYT (Depth 8)	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
TextRCNN (Zhou et al., 2020)	83.55	76.99	81.57	59.25	70.83	56.18
HiAGM (Zhou et al., 2020)	85.82	80.28	83.96	63.35	74.97	60.83
HTCInfoMax (Deng et al., 2021)	85.58	80.05	83.51	62.71	-	-
HiMatch (Chen et al., 2021)	86.20	80.53	84.73	64.11	-	-
BERT (Wang et al., 2022)	85.63	79.07	85.65	67.02	78.24	66.08
BERT+HiAGM(Wang et al., 2022)	86.04	80.19	85.58	67.93	78.64	66.76
BERT+HTCInfoMax(Wang et al., 2022)	86.30	79.97	85.53	67.09	78.75	67.31
BERT+HiMatch (Chen et al., 2021)	86.70	81.06	86.33	68.66	-	-
HGCLR (Wang et al., 2022)	87.11	81.20	86.49	68.31	78.86	67.96
BERT+HardPrompt (Ours)	86.39	80.43	86.78	68.78	79.45	67.99
BERT+SoftPrompt (Ours)	86.57	80.75	86.53	68.34	78.95	68.21
HPT (Ours)	**87.16**	**81.93**	**87.26**	**69.53**	**80.42**	**70.42**

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

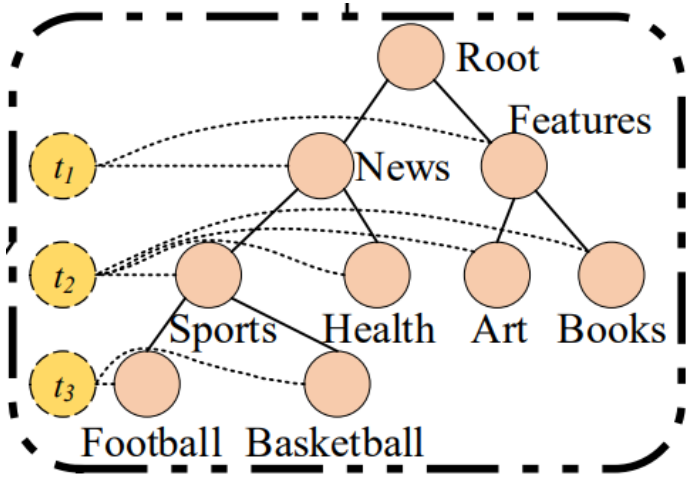
Experiments

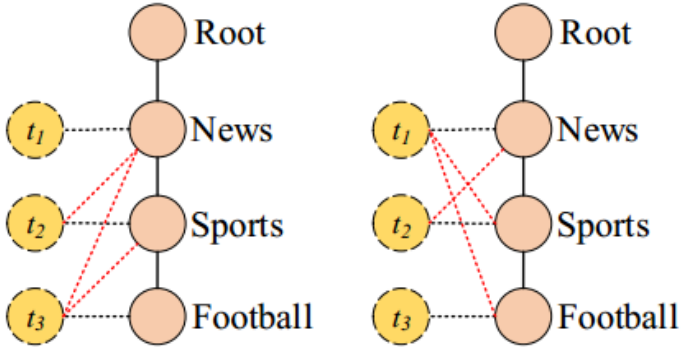
Ablation Models	Micro-F1	Macro-F1
HPT	**80.49**	**71.07**
r.m. hierarchy constraint	80.32	70.58
r.m. hierarchy injection	80.41	69.71
r.p. BCE loss	79.74	70.40
r.m. MLM loss	80.16	70.78
with random connection	80.12	69.42

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

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

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





(a) Depth increasing (b) Random connection

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

Experiments

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

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

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

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

Experiments

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

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

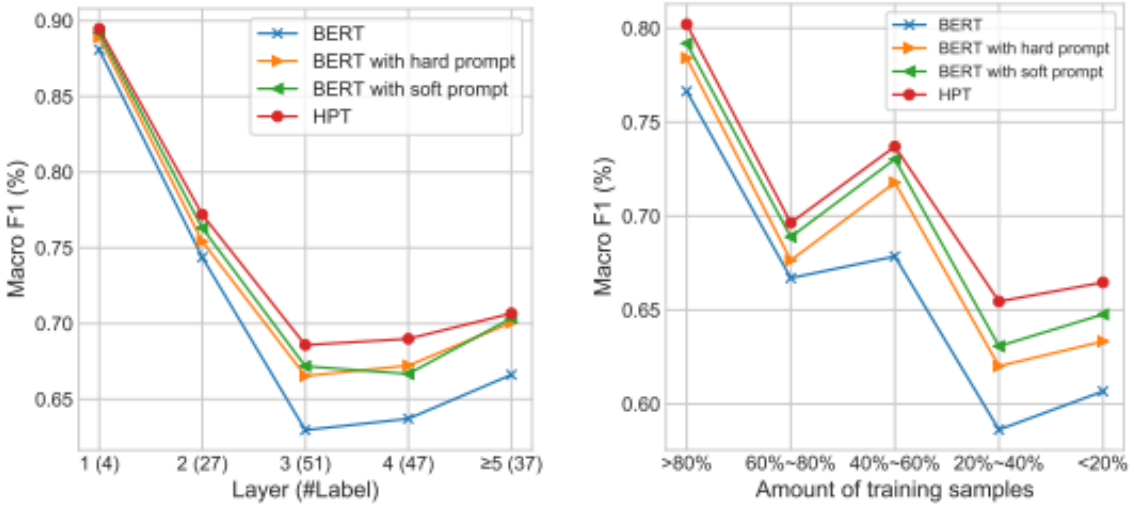


Figure 3: Macro F1 scores of label clusters on the development set of NYT. (a) Label clusters grouped by depth in the hierarchy. (b) Label clusters grouped by amount of training samples. >80% represents cluster of top 20% labels ranking by amount of training samples. The rest clusters are arranged similarly.

Experiments

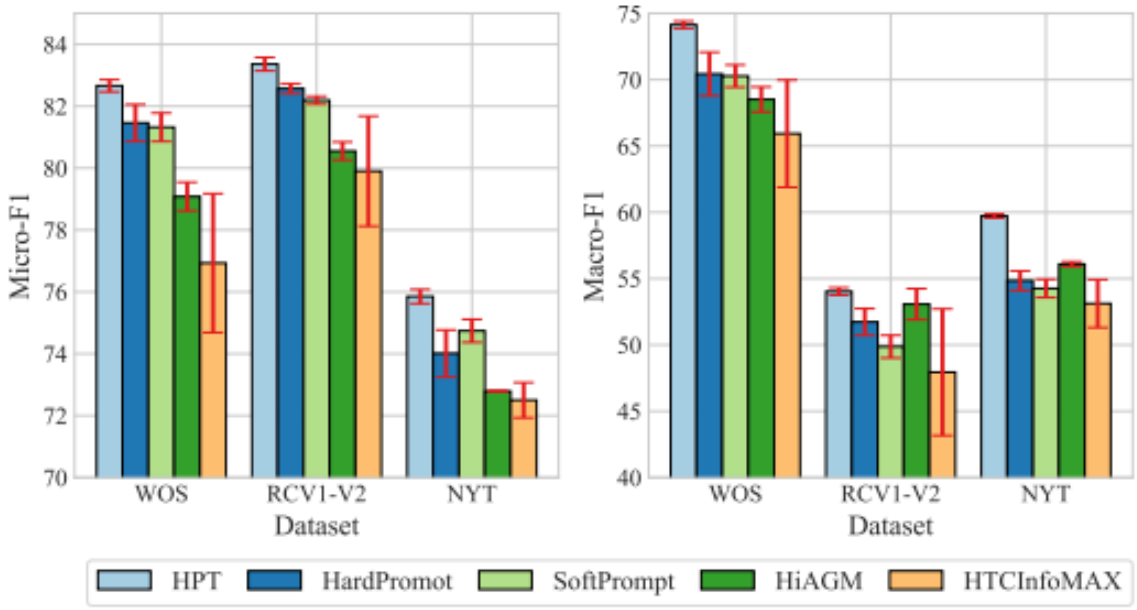


Figure 4: F1 scores on 3 mini training dataset with only 10% training instances of the full training dataset. We report the average scores with standard deviation over 3 different runs.

Thank you !