Domain-Hierarchy Adaptation via Chain of Iterative Reasoning for Few-shot Hierarchical Text Classification

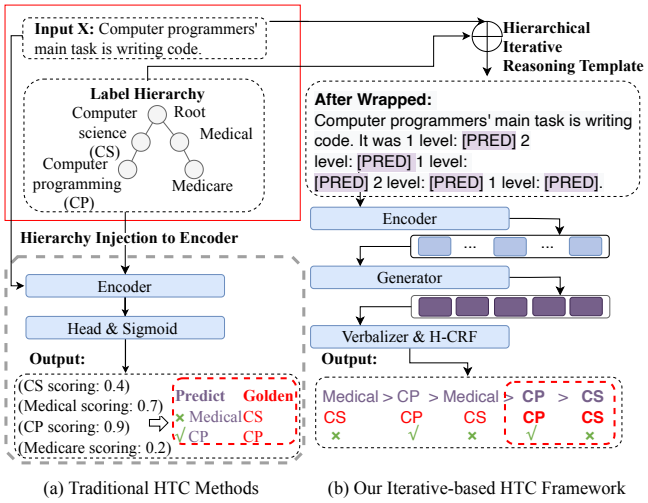
Ke Ji[1] , **Peng Wang**[1,2*] , **Wenjun Ke**[1,2] , **Guozheng Li**[1] ,
Jiajun Liu[1] , **Jingsheng Gao**[3] , **Ziyu Shang**[1]
[1]School of Computer Science and Engineering, Southeast University
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education
[3]School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University
{keji, pwang, kewenjun, gzli, jiajliu, ziyus1999}@seu.edu.cn, gaojingsheng@sjtu.edu.cn

Abstract

Recently, various pre-trained language models (PLMs) have been proposed to prove their impressive performances on a wide range of few-shot tasks. However, limited by the unstructured prior knowledge in PLMs, it is difficult to maintain consistent performance on complex structured scenarios, such as hierarchical text classification (HTC), especially when the downstream data is extremely scarce. The main challenge is how to transfer the unstructured semantic space in PLMs to the downstream domain hierarchy. Unlike previous work on HTC which directly performs multi-label classification or uses graph neural network (GNN) to inject label hierarchy, in this work, we study the HTC problem under a few-shot setting to adapt knowledge in PLMs from an unstructured manner to the downstream hierarchy. Technically, we design a simple yet effective method named Hierarchical Iterative Conditional Random Field (HierICRF) to search the most domain-challenging directions and exquisitely crafts domain-hierarchy adaptation as a hierarchical iterative language modeling problem, and then it encourages the model to make hierarchical consistency self-correction during the inference, thereby achieving knowledge transfer with hierarchical consistency preservation. We perform HierICRF on various architectures, and extensive experiments on two popular HTC datasets demonstrate that prompt with HierICRF significantly boosts the few-shot HTC performance with an average Micro-F1 by 28.80% to 1.50% and Macro-F1 by 36.29% to 1.5% over the previous state-of-the-art (SOTA) baselines under few-shot settings, while remaining SOTA hierarchical consistency performance.

1 Introduction

Pre-trained Language Models (PLMs) [Radford *et al.*, 2018; Devlin *et al.*, 2019; Raffel *et al.*, 2020] have gained significant prominence for their exceptional performance across various language-related tasks, including text classification [Kowsari *et al.*, 2019], and relation extraction [Wang *et al.*, 2023; Li *et al.*, 2022].The success of PLMs mainly benefits from large-scale pre-training and sufficient downstream labeled data. However, when downstream labeled data is scarce, performance is greatly compromised, and it is further aggravated when we try to transfer unstructured prior knowledge in PLMs to downstream structured tasks like hierarchical text classification (HTC) [Mao *et al.*, 2019]. Due to HTC's broad range of practical applications [Mao *et al.*, 2019], including product categorization [Cevahir and Murakami, 2016], fine-grained entity typing [Xu and Barbosa, 2018] and news classification [Irsan and Khodra, 2019], HTC has remained a significant research challenge over time. Despite existing HTC methods, its complex label hierarchy and the need for extensive annotation still hinder performance in practice. Addressing HTC in few-shot



(a) Traditional HTC Methods (b) Our Iterative-based HTC Framework

Figure 1: Illustration of methods for HTC. The red sequence represents the golden label, and the purple sequence represents the predicted sequence. Hierarchical inconsistency happens when the relationship between the outputs of different layers conflicts with the hierarchical dependency tree, for example, the model predicts the CP which is not the child node of its other output like Medicare.

*Corresponding author

scenarios remains an open area of research [Ji *et al.*, 2023].

Currently, the state-of-the-art in HTC [Wang *et al.*, 2022a; Wang *et al.*, 2022b; Ji *et al.*, 2023] involves incorporating label hierarchy features into the input or output layer of a text encoder, using graph encoders or hierarchical verbalizer, as is illustrated in Figure 1(a). It's disappointing that they ignore the adaptation of unstructured prior knowledge to downstream domain-hierarchy structure and still directly consider HTC as a multi-label classification problem based on an encoder with label-hierarchy dependencies injection. Inspired by the *in-context learning* approach proposed by GPT-3 [Brown *et al.*, 2020] and prompt-based methods [Petroni *et al.*, 2019; Gao *et al.*, 2021; Schick and Schütze, 2021; Li *et al.*, 2023; Li *et al.*, 2024] that aim to bridge the gap between pre-training and downstream tasks by utilizing few hard or soft prompts to stimulate the PLMs' knowledge, [Wang *et al.*, 2022b; Ji *et al.*, 2023] are proposed to provide a more systematic study under low resource or few-shot settings using prompt-based methods. However, instead of paying more attention to the inherent difference between downstream hierarchy and unstructured objective in PLMs, all these methods are just built from the perspective of how to get a hierarchical label dependency representation, leading to unstable hierarchical consistency performance. Similar to mathematical reasoning tasks [Zhu *et al.*, 2023] for which the answers are often implicit, making it difficult to deal with directly through question-answer pairs, thus the way to think about the HTC task is really unfriendly and even antithetical to the language model's capabilities. Thus the primary difference between these works on HTC tasks is the way they inject the label hierarchy constraint.

Despite their success, they still suffer from two limitations. On the one hand, previous methods are not well designed to focus on how to decompose and simplify hierarchy-based problems, on the contrary, they think of HTC in an even more complex way, and thus they are difficult to handle the hierarchical inconsistency problem. On the other hand, considering most of the currently popular large language models are based on encoder-decoder [Raffel *et al.*, 2020; Chung *et al.*, 2022] or decoder-only [Brown *et al.*, 2020; Scao *et al.*, 2022] architectures, previous works are mostly applicable to the encoder-only architecture, which leads them to mine rich prior knowledge at a limited model scale in practical applications. Therefore, few studies have investigated how to efficiently handle the few-shot domain-hierarchy adaptation problem. And few studies have tried to develop a simple and unified framework that can be flexibly deployed in any architectural model for better practical application performance.

In this work, we design a unified framework named HierICRF from the perspective of path routing that can be deployed on any transformer-based architecture to fully elicit the potential of unstructured prior knowledge in PLMs to complete downstream hierarchy tasks. Unlike previous works that mainly focused on how to align their carefully crafted representators that incorporate label-hierarchy dependencies with the sentence semantic space, we use a language modeling routing paradigm based on hierarchical iteration to unify the objectives of the two stages of language modeling in pre-training and downstream hierarchy-based tasks, which is more feasible. Technically, as is shown in Figure 1(b), (1) Firstly, we construct a hierarchy-aware prompt to encourage the model to generate hierarchically repeated series. (2) Secondly, this series will be fed into a verbalizer to obtain their masked language modeling (MLM) logits of labels in the hierarchical dependency tree. (3) Finally, we use a hierarchical iterative CRF and initialize its transition matrix (e.g., transition scores between non-adjacent layers are set to a minimum to avoid erroneous cross-layer transfers) based on the hierarchical dependency tree to constrain the hierarchical dependency during the path routing process. Combining these three stages, with the deepening of the hierarchically repeated reasoning process, the model can perform hierarchical consistency self-correction during each step to encourage predictions more accurate. During the inference stage, we use the Viterbi algorithm [Forney, 1973] to decode the series to obtain our final predictions.

The main contributions of this paper are summarized as:

- To our best knowledge, we are among the first to investigate a few-shot HTC framework that emphasizes domain-hierarchy adaptation to bridge the gap between unstructured prior knowledge and downstream hierarchy.
- We proposed a unified framework that is suitable for any transformer-based architecture to efficiently mine prior knowledge within limited downstream labeled datasets for better few-shot learning.
- We thoroughly study the hierarchical inconsistency problem. Experiments on BERT and T5 demonstrate that HierICRF outperforms the previous SOTA few-shot HTC methods on two popular datasets under extreme few-shot settings while achieving SOTA hierarchical consistency performance with an average of 9.3% and 4.38% CMacro-F1 improvements on WOS and DBpedia, respectively.

2 Related Work

Hierarchical Text Classification. Current HTC research mainly focuses on how to incorporate hierarchical label knowledge to address imbalanced and large-scale label hierarchy challenges [Mao *et al.*, 2019]. Various approaches have been explored, including label-based attention modules [Zhang *et al.*, 2022], meta-learning [Wu *et al.*, 2019], and reinforcement learning methods [Mao *et al.*, 2019]. [Zhou *et al.*, 2020] proposes a more holistic approach named HiAGM by encoding the entire label structure with hierarchy encoders, which has shown greater performance improvements. Recent works [Wang *et al.*, 2021; Chen *et al.*, 2021] have also explored matching learning and concept enrichment to exploit the relationship between text and label semantics. Later works such as HGCLR [Wang *et al.*, 2022a] and HPT [Wang *et al.*, 2022b] have migrated the label hierarchy into text encoding, achieving excellent performances through prompt tuning methods. [Ji *et al.*, 2023] proposes a multiple verbalizers framework to reduce the gap between PLMs and HTC for better few-shot learning. Despite these advances, how to model HTC tasks with a unified hierarchy-aware paradigm is still underexplored, and there is a need to design a solution from the perspective of the path that performs well on consistency performance within limited training samples.

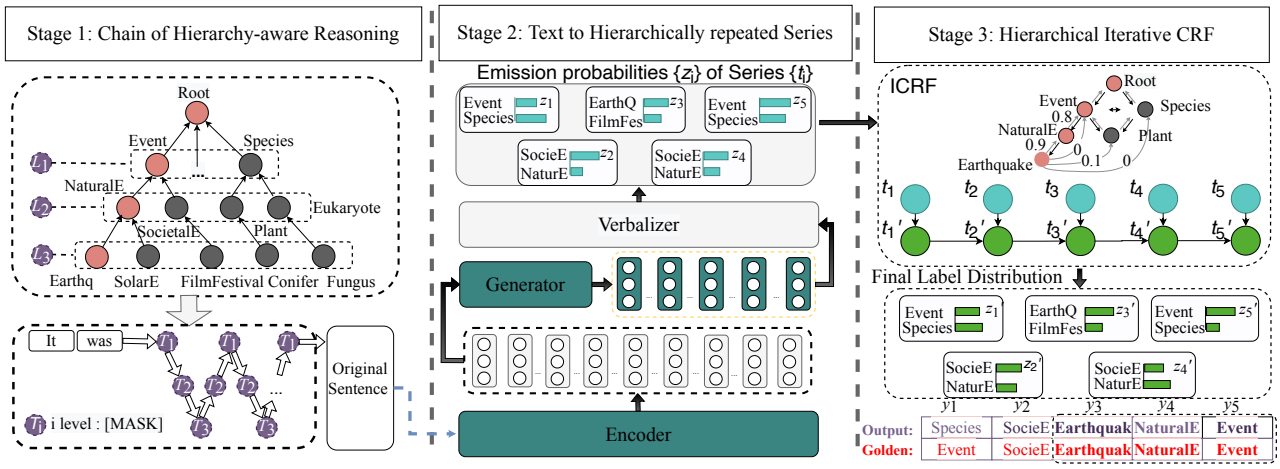


Figure 2: The overview of HierICRF. There are two ways to inject hierarchical constraints: (a) Chain of hierarchy-aware reasoning and (b) Hierarchical iterative CRF. At stage 3, we select the predictions $\{y_3, y_4, y_5\}$ (the last path routing iteration) as the final outputs.

3 Methodology

3.1 Problem Statements

This paper mainly focuses on few-shot HTC tasks. Different from previous works that assume a rich labeled dataset, we adopt the few-shot setting, i.e., only a limited number of samples are available for fine-tuning which is more practical as it assumes minimal resources. We follow the greedy sampling algorithm proposed in [Ji *et al.*, 2023] to obtain the support set *S* from the original HTC datasets, ensuring that each label path has exactly *K*-shot samples. Formally, the K-shot HTC task is defined as follows: given a text x and a K-shot support set *S* for the target mandatory-leaf path set $C_{\mathcal{T}}$, the goal is to predict all golden paths on the label hierarchy tree, where $C_{\mathcal{T}}$ consists of all paths from the root node to the leaf nodes on the label hierarchy tree $\mathcal{H}$. In our settings, there is only one correct label for each layer.

3.2 Framework Overview

Figure 2 shows the overall architecture of our proposed HierICRF. Since the label structure that this paper focuses on is a tree, we view the HTC task as a process of a chain of thoughts via iterative CRF. Specifically, in the first stage, we construct a hierarchy-aware reasoning chain T_{chain} as our prompt to deepen the process of hierarchy-aware reasoning. During the second stage, we use T_{chain} to guide the generator to generate hierarchically repeated series $\boldsymbol{t}$, the $\boldsymbol{t}$ will be fed into a verbalizer to obtain their MLM logits z in the label hierarchy tree. Finally, during layer-to-layer transfer over the MLM logits z_i of the series $\boldsymbol{t}_i$, we use a hierarchical iterative CRF (ICRF) to model its path routing by incorporating hierarchical information into the transition matrix of ICRF.

3.3 Chain of Hierarchy-Aware Reasoning

Due to the complex label dependency in HTC task, it is hard to adapt flat prior knowledge in PLMs to downstream hierarchical tasks. Besides, hierarchical dependency information between labels at different layers or the same layers may be implicit. To address the above issues, inspired by the chain of thought (COT) [Wei *et al.*, 2022] and math [Zhu *et al.*, 2023] that solve mathematical or planning-related problems by decomposing complex problems into sub-processes, we propose a simple but effective method named hierarchy-aware reasoning chain to elegantly flattens HTC into a hierarchical loop-based language modeling process, thus allowing it to fully exploit the capabilities of PLM. The details of obtaining the prompt of the hierarchy-aware reasoning chain named T_{chain} are shown in Algorithm 1. For example, when the depth of the hierarchy tree is 2 and the number of iterations T_{chain} is 2, the reasoning chain template T_{chain} is simply like "x. It was 1 level: [MASK] 2 level: [MASK] 1 level: [MASK] 2 level: [MASK] 1 level: [MASK]. ". The T_{chain} will later be used to guide the model to generate hierarchically repeated series.

3.4 Text to Hierarchically Repeated Series

To verify the ability of our method for few-shot domain-hierarchy adaptation, we implement our method on both BERT and T5. We first feed the input text x wrapped with template T_{chain} into the encoder to obtain the hidden states $\boldsymbol{h}_{1:n}$:

$$\boldsymbol{h}_{1:n} = \text{Encoder}(T_{chain}(x)_{1:n}) \quad (1)$$

where $\boldsymbol{h}_{1:n} \in \mathbb{R}^{n \times r}$, and r denotes the hidden state dimension of encoder and n represents the length of $T_{chain}(x)$. We then obtain target hierarchically repeated series $\boldsymbol{t}_{1:l} = \{\boldsymbol{t}_i\}$ through:

$$t_{1:l} = \text{Generator}(\boldsymbol{h}_{1:n}) \quad (2)$$

For encoder-only LMs: Generator$(\cdot)$ means to directly extract a subset $\{\boldsymbol{h}_j\}$ consisting of hidden state vectors corresponding to all `[MASK]` tokens from $\boldsymbol{h}_{1:n}$ as our $\boldsymbol{t}_{1:l}$. *For encoder-decoder LMs*: Generator$(\cdot)$ represents feeding $\boldsymbol{h}_{1:n}$ into and prompt its decoder to obtain our final $\boldsymbol{t}_{1:l}$.

Furthermore, we construct a flat-verbalizer *V* based on all labels on the hierarchical tree for label mapping learning. The verbalizer is represented as a continuous vector $\boldsymbol{W}_V \in \mathbb{R}^{r \times m}$, where m signifies the number of labels. The embedding $\boldsymbol{W}_V$ is initialized by averaging the embeddings of its corresponding label tokens. We feed the series $\{\boldsymbol{t}_i\}$ into the verbalizer to get the emission probabilities $z = \{z_1, ..., z_l\}$.

Algorithm 1 Generate hierarchy-aware reasoning chain

Input: number of iterations I_{chain}, label hierarchy $\mathcal{H}$, depth of the $\mathcal{H}$ called $\mathcal{D}$
Output: hierarchy reasoning chain

$T_{chain} \leftarrow$ "It was" //Initialize the prompt
for $i = 1$ *to* $|\mathcal{D}|$ **do**
  T_{chain} = T_{chain} + "i level: `[MASK]`"
end for
for $i = \mathcal{D} - 1$ *to* 1 **do**
  T_{chain} = T_{chain} + "i level: `[MASK]`"
end for
for $\_ = 1$ *to* $|I_{chain} - 1|$ **do**
  for $i = \mathcal{D}$ *to* 1 **do**
    T_{chain} = T_{chain} + "i level: `[MASK]`"
  end for
end for
return T_{chain} =0

3.5 Hierarchical Iterative Conditional Random Fields

After obtaining the label distribution of each step in the hierarchy-aware reasoning chain, instead of directly classifying, we regard this hierarchically repeated series as a process of hierarchical path routing step-by-step. Inspired by CRF [Sutton *et al.*, 2012] widely used to model the transition of state space of time series in named entity recognition [Nadeau and Sekine, 2007], here we model this sequencing process using hierarchical iterative CRF, injecting hierarchical constraint by optimizing transition matrices.

Formally, given a sentence *x*, we use $z = \{z_1, ..., z_l\}$ to represent a generic series where z_i is the emission probability of the *i*-th routing step. $y = \{y_1, ..., y_n\}$ represents the golden labels for z. The probabilistic model of a sequence CRF defines a set of conditional probabilities $p(y|z; \boldsymbol{W}, b)$ over all possible label sequences y given z as:

$$p(y|z; W, b) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, z)}{\sum_{y' \in \mathcal{H}} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y'_i, z)} \tag{3}$$

where $\psi_i(y', y, z) = \exp(\boldsymbol{W}_{y',y}^T z_i + b_{y',y})$ are potential functions, and $\boldsymbol{W}_{y',y}^T$ and $b_{y',y}$ denote the weight and bias corresponding to label pair (y', y), respectively. The final training objective is calculated as:

$$L(\boldsymbol{W}, b) = \sum_i log\ p(y|z; \boldsymbol{W}, b) \tag{4}$$

Additionally, we initialize the transition scores between non-adjacent layers to a minimum to avoid erroneous cross-layer transfers. The transition score between Earthquake and Species in Figure 2 is initialized to 0 before training.

3.6 Decoding

After training, the decoding is to search for the label sequence y^* with the maximum conditional probability:

$$y^* = \arg\max\ p(y|x; \boldsymbol{W}, b); y \in \mathcal{H} \tag{5}$$

where we pick out the outputs of the last path routing iteration from y^* as our final predictions.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation Metrics. We evaluate our method and all baselines on two popular datasets for HTC: Web-of-Science (WOS) [Kowsari *et al.*, 2017], DBpedia [Sinha *et al.*, 2018]. WOS is a database that includes abstracts of published papers, among other bibliographic information such as author names, journal titles, and publication dates. DBpedia is a bigger dataset with labels from Wikipedia meta information provider DBpedia with a three-level hierarchy. Table 1 presents the statistical details. There are differences in the domain distribution of these two datasets.

We measure the experimental results with Macro-F1 and Micro-F1. To more thoroughly assess the hierarchical consistency, we utilize the path-constrained C-metric proposed in [Yu *et al.*, 2022] and the path-based P-metric proposed in [Ji *et al.*, 2023]. The C-metric only considers a correct prediction for a label node to be valid if all of its ancestor nodes are predicted correctly. In contrast, the P-metric requires that all of the ancestors and child nodes on the path to which the label node belongs are predicted correctly for its correct prediction to be considered valid.

Baselines. For performance comparison of various models from a different perspective, we select the following strong baselines: HiMatch-BERT [Chen *et al.*, 2021], HGCLR [Wang *et al.*, 2022a], HPT [Wang *et al.*, 2022b] and HierVerb [Ji *et al.*, 2023]. We also perform Vanilla Fine-Tuning (Vanilla FT) [Devlin *et al.*, 2019] and Vanilla Soft Verbalizer (SoftVerb) [Schick and Schütze, 2021] method on the Few-shot HTC task. Vanilla FT is a simple method consisting of a Binary CrossEntropy loss for ordinary multi-label classification followed by a classifier. SoftVerb uses the traditional template "x. It was 1:level [MASK] 2:level [MASK].", then the hidden states of all positions are fed into the verbalizer to obtain label logits. Note that SoftVerb, HPT, and HierVerb are prompt-based methods and all baselines above are limited to encoder-only architectures. Considering that the current research based on encoder-decoder performs poorly under few-shot scenario, we construct a strong baseline called SoftVerb-T5 by removing all components on HierICRF-T5 in the ablation experiment for a fair comparison.

Backbone and Implementation Details. We adopt both BERT (`BERT-base-uncased`) [Devlin *et al.*, 2019] and T5 (`T5-base`) [Raffel *et al.*, 2020] as the main backbone of our experiments, for additional experiments we use BERT (`BERT-base-uncased`) by default.

Datasets	DBpedia	WOS
Level 1 Categories	9	7
Level 2 Categories	70	134
Level 3 Categories	219	NA
Number of documents	381025	46985
Mean document length	106.9	200.7

Table 1: Comparison of popular HTC datasets.

K	Method	WOS(Depth 2)		DBpedia(Depth 3)	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1
1	BERT (Vanilla FT)	2.99 ± 20.85 (5.12)	0.16 ± 0.10 (0.24)	14.43 ± 13.34 (24.27)	0.29 ± 0.01 (0.32)
	HiMatch-BERT [Chen *et al.*, 2021]	43.44 ± 8.90 (48.26)	7.71 ± 4.90 (9.32)	-	-
	HGCLR [Wang *et al.*, 2022a]	9.77 ± 11.77 (16.32)	0.59 ± 0.10 (0.63)	15.73 ± 31.07 (25.13)	0.28 ± 0.10 (0.31)
	HPT [Wang *et al.*, 2022b]	50.05 ± 6.80 (50.96)	25.69 ± 3.31 (27.76)	72.52 ± 10.20 (73.47)	31.01 ± 2.61 (32.50)
	SoftVerb [Schick and Schütze, 2021]	56.11 ± 7.44 (58.13)	41.35 ± 5.62 (44.32)	90.38 ± 0.10 (90.89)	82.72 ± 0.18 (83.74)
	HierVerb [Ji *et al.*, 2023]	58.95 ± 6.38 (61.76)	44.96 ± 4.86 (48.19)	91.81 ± 0.07 (91.95)	85.32 ± 0.04 (85.44)
	HierICRF-BERT (Ours)	**59.40 ± 6.22 (62.01)**	46.49 ± 3.91 (49.54)	**92.05 ± 0.10 (92.11)**	86.10 ± 0.10 (86.78)
	HierICRF-T5 (Ours)	59.20 ± 5.17 (61.97)	**47.72 ± 2.23 (50.22)**	91.94 ± 0.05 (92.01)	**86.70 ± 0.05 (86.94)**
2	BERT (Vanilla FT)	46.31 ± 0.65 (46.85)	5.11 ± 1.31 (5.51)	87.02 ± 3.89 (88.20)	69.05 ± 26.81 (73.28)
	HiMatch-BERT [Chen *et al.*, 2021]	46.41 ± 1.31 (47.23)	18.97 ± 0.65 (21.06)	-	-
	HGCLR [Wang *et al.*, 2022a]	45.11 ± 5.02 (47.56)	5.80 ± 11.63 (9.63)	87.79 ± 0.40 (88.42)	71.46 ± 0.17 (71.78)
	HPT [Wang *et al.*, 2022b]	57.45 ± 1.89 (58.99)	35.97 ± 11.89 (39.94)	90.32 ± 0.64 (91.11)	81.12 ± 1.33 (82.42)
	SoftVerb [Schick and Schütze, 2021]	62.31 ± 13.24 (65.02)	49.33 ± 6.55 (53.46)	92.97 ± 0.20 (93.04)	87.61 ± 0.20 (87.87)
	HierVerb [Ji *et al.*, 2023]	**66.08 ± 4.19 (68.01)**	54.04 ± 3.24 (56.69)	93.71 ± 0.01 (93.87)	88.96 ± 0.02 (89.02)
	HierICRF-BERT (Ours)	65.71 ± 3.69 (67.87)	55.18 ± 3.11 (57.12)	**94.22 ± 0.01 (94.22)**	**89.31 ± 0.01** (89.31)
	HierICRF-T5 (Ours)	65.52 ± 2.43 (66.97)	**56.11 ± 1.79 (57.49)**	93.64 ± 0.01 (94.01)	89.22 ± 0.01 (**89.45**)
4	BERT (Vanilla FT)	56.00 ± 4.25 (57.18)	31.04 ± 16.65 (33.77)	92.94 ± 0.66 (93.38)	84.63 ± 0.17 (85.47)
	HiMatch-BERT [Chen *et al.*, 2021]	57.43 ± 0.01 (57.43)	39.04 ± 0.01 (39.04)	-	-
	HGCLR [Wang *et al.*, 2022a]	56.80 ± 4.24 (57.96)	32.34 ± 15.39 (33.76)	93.14 ± 0.01 (93.22)	84.74 ± 0.11 (85.11)
	HPT [Wang *et al.*, 2022b]	65.57 ± 1.69 (67.06)	45.89 ± 9.78 (49.42)	94.34 ± 0.28 (94.83)	90.09 ± 0.87 (91.12)
	SoftVerb [Schick and Schütze, 2021]	69.58 ± 3.27 (71.00)	58.53 ± 1.64 (60.18)	94.47 ± 0.10 (94.74)	90.25 ± 0.10 (90.73)
	HierVerb [Ji *et al.*, 2023]	72.58 ± 0.83 (73.64)	63.12 ± 1.48 (64.47)	94.75 ± 0.13 (95.13)	90.77 ± 0.33 (91.43)
	HierICRF-BERT (Ours)	**73.83 ± 0.71 (74.19)**	65.40 ± 0.69 (65.86)	**95.14 ± 0.15 (95.20)**	**91.20 ± 0.05 (91.81)**
	HierICRF-T5 (Ours)	73.22 ± 0.36 (60.11)	**65.61 ± 0.54 (66.12)**	94.66 ± 0.01 (95.10)	90.89 ± 0.01 (91.33)
8	BERT (Vanilla FT)	66.24 ± 1.96 (67.53)	50.21 ± 5.05 (52.60)	94.39 ± 0.06 (94.57)	87.63 ± 0.28 (87.78)
	HiMatch-BERT [Chen *et al.*, 2021]	69.92 ± 0.01 (70.23)	57.47 ± 0.01 (57.78)	-	-
	HGCLR [Wang *et al.*, 2022a]	68.34 ± 0.96 (69.22)	54.41 ± 2.97 (55.99)	94.70 ± 0.05 (94.94)	88.04 ± 0.25 (88.61)
	HPT [Wang *et al.*, 2022b]	76.22 ± 0.99 (77.23)	67.20 ± 1.89 (68.63)	95.49 ± 0.01 (95.57)	92.35 ± 0.03 (92.52)
	SoftVerb [Schick and Schütze, 2021]	75.99 ± 0.47 (76.77)	66.99 ± 0.27 (67.50)	95.48 ± 0.01 (95.64)	92.06 ± 0.01 (92.37)
	HierVerb [Ji *et al.*, 2023]	78.12 ± 0.55 (**78.87**)	69.98 ± 0.91 (71.04)	95.69 ± 0.01 (95.70)	92.44 ± 0.01 (92.51)
	HierICRF-BERT (Ours)	**78.54 ± 0.25** (78.69)	70.79 ± 0.38 (71.35)	**95.80 ± 0.01 (95.85)**	**92.77 ± 0.01** (92.82)
	HierICRF-T5 (Ours)	77.78 ± 0.17 (78.64)	**71.62 ± 0.10 (71.90)**	95.55 ± 0.01 (95.70)	92.69 ± 0.01 (92.79)
16	BERT (Vanilla FT)	75.52 ± 0.32 (76.07)	65.85 ± 1.28 (66.96)	95.31 ± 0.01 (95.37)	89.16 ± 0.07 (89.35)
	HiMatch-BERT [Chen *et al.*, 2021]	77.67 ± 0.01 (78.24)	68.70 ± 0.01 (69.58)	-	-
	HGCLR [Wang *et al.*, 2022a]	76.93 ± 0.52 (77.46)	67.92 ± 1.21 (68.66)	95.49 ± 0.04 (95.63)	89.41 ± 0.09 (89.71)
	HPT [Wang *et al.*, 2022b]	79.85 ± 0.41 (80.58)	72.02 ± 1.40 (73.31)	96.13 ± 0.01 (96.21)	93.34 ± 0.02 (93.45)
	SoftVerb [Schick and Schütze, 2021]	79.62 ± 0.85 (80.68)	70.95 ± 0.62 (71.84)	95.94 ± 0.15 (96.18)	92.89 ± 0.20 (93.37)
	HierVerb [Ji *et al.*, 2023]	80.93 ± 0.10 (**81.26**)	73.80 ± 0.12 (74.19)	96.17 ± 0.01 (96.21)	93.28 ± 0.06 (93.49)
	HierICRF-BERT (Ours)	**81.02 ± 0.10** (81.20)	74.05 ± 0.10 (74.15)	**96.22 ± 0.01 (96.25)**	93.38 ± 0.02 (93.60)
	HierICRF-T5 (Ours)	80.94 ± 0.05 (81.05)	**75.23 ± 0.05 (75.88)**	96.11 ± 0.01 (95.85)	**93.56 ± 0.01 (93.70)**

Table 2: Results of 1/2/4/8/16-shot HTC. F1 scores on WOS and DBpedia. We report the mean F1 scores (%) over 3 random seeds. Bold: best results. Underlined: second highest. For baseline models, we report the F1 scores from their original paper.

The batch size is 8. We use a learning rate of 5e-5 for BERT and 3e-5 for T5 and train the model for 20 epochs. For soft verbalizer and ICRF, we use a learning rate of 1e-4 to encourage a faster convergence. After each epoch, we evaluate the model's performance on the development set and set early stopping to 5 as usual. For the baseline models, we follow the hyperparameter set as specified in their respective papers.

4.2 Main Results

Main experimental results are reported in Table 2. By optimizing from a path routing perspective, HierICRF outperforms overall comparison baselines under most situations.

We first find out that prompt-based methods outperform vanilla FT by a dramatic margin in the case of no more than 4-shot settings. On average, 67.01%, 13.30%, and 10.01% Micro-F1 improvements are achieved by HierICRF-BERT compared to the vanilla FT. However, although HPT is designed as a prompt-based method, its few-shot results are not satisfactory. The reason is obvious since the overfitting problem of the GNN layers becomes serious when labeled data used for fine-tuning is limited.

Second, HierICRF-BERT achieves 1.53%, 1.14%, and 2.28% Macro-F1 improvements from the best baselines on 1, 2, and 4-shot on WOS, respectively. Besides, HierICRF-T5 has a substantial improvement with an average Macro-F1 of 2.08% on WOS and 0.46% on DBpedia compared to the previous SOTA HierVerb while keeping competitive with HierICRF-BERT on Micro-F1.

Additionally, it is clear that both the HierICRF-BERT and

K	Method	WOS				DBpedia			
		PMicro-F1	PMacro-F1	CMicro-F1	CMacro-F1	PMicro-F1	PMacro-F1	CMicro-F1	CMacro-F1
1	HierICRF-BERT	**42.63**	**41.62**	**57.35**	**44.06**	**84.95**	**79.44**	**90.52**	**84.42**
	HierVerb	39.77	37.24	55.18	39.42	83.56	77.96	89.80	81.78
	HPT	19.97	17.47	49.10	22.92	61.08	57.80	82.84	66.99
	HGCLR	0.0	0.0	2.21	0.09	0.0	0.0	28.05	0.24
	Vanilla FT	0.0	0.0	0.96	0.04	0.0	0.0	28.08	0.24
2	HierICRF	**52.62**	**49.39**	**64.07**	**52.64**	**90.14**	**87.66**	**94.54**	**90.26**
	HierVerb	50.15	47.98	62.90	49.67	88.58	86.35	93.61	88.96
	HPT	28.27	26.51	56.64	33.50	82.36	81.41	92.31	86.43
	HGCLR	1.39	1.49	45.01	4.88	54.55	3.72	67.70	26.41
	Vanilla FT	1.43	1.42	45.75	4.95	53.83	3.71	67.72	26.89
4	HierICRF	**63.91**	**60.17**	**73.45**	**62.88**	**93.20**	**92.51**	**95.88**	**93.31**
	HierVerb	62.16	59.70	72.41	61.19	91.90	91.38	95.74	92.87
	HPT	50.96	48.76	69.43	55.27	87.61	87.04	94.50	90.42
	HGCLR	29.94	27.70	57.43	34.03	55.34	3.76	67.54	28.60
	Vanilla FT	22.97	20.73	55.10	27.50	55.15	3.74	67.44	28.32

Table 3: Consistency experiments on the WOS and DBpedia datasets using two path-constraint metrics. PMicro-F1 and PMacro-F1 are our proposed path-based consistency evaluation P-metric. We report the mean F1 scores (%) over 3 random seeds. All experiments use their respective metrics as a signal for early stopping.

K	Ablation Models	BERT		T5	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1
1	Ours	**59.40**	**46.49**	**59.20**	**47.72**
	$r.m.$ ICRF loss	57.54	43.25	57.82	45.77
	$r.m.$ CHR	57.82	43.71	56.88	44.25
	$r.m.$ ICRF&CHR	56.11	41.35	57.16	44.83
2	Ours	**65.71**	**55.18**	**65.52**	**56.11**
	$r.m.$ ICRF loss	64.11	52.05	64.29	53.47
	$r.m.$ CHR	64.52	52.39	64.73	54.82
	$r.m.$ ICRF&CHR	62.31	49.33	62.07	49.67
4	Ours	**73.83**	**65.40**	**73.22**	**65.61**
	$r.m.$ ICRF loss	71.78	63.99	71.21	63.48
	$r.m.$ CHR	71.51	63.29	72.52	64.15
	$r.m.$ ICRF&CHR	69.58	58.83	70.24	59.75
8	Ours	**78.54**	**70.79**	**77.78**	**71.62**
	$r.m.$ ICRF loss	76.44	68.23	76.52	69.30
	$r.m.$ CHR	77.29	69.60	77.63	70.44
	$r.m.$ ICRF&CHR	75.99	66.99	75.24	67.11
16	Ours	**81.02**	**74.05**	**80.94**	**75.23**
	$r.m.$ ICRF loss	80.04	72.71	80.15	72.41
	$r.m.$ CHR	80.54	73.64	80.46	73.84
	$r.m.$ ICRF&CHR	79.62	70.95	78.91	71.74

Table 4: Ablation experiments on WOS. $r.m.$ stands for $remove$. We report the mean F1 scores (%) over 3 random seeds. ICRF stands for Hierarchical Iterative CRF while CHR stands for Chain of Hierarchy-aware Reasoning. Note when all mechanisms are removed, HierICRF is equivalent to Vanilla SoftVerb.

HierICRF-T5's Micro-F1 and Macro-F1 change very slightly from 1 to 16 shots on DBpedia while other models except HierVerb are particularly dependent on the increase of labeled training samples. For example, as the shots become fewer, the HierICRF-BERT's Micro-F1 changes from 96.22% to 92.05% while HGCLR's Micro-F1 changes from 95.49% to 15.73%. The results indicate that our method can efficiently mine the prior knowledge in pre-training for hierarchical tasks in the case of extremely few samples instead of relying too much on limited training samples to optimize the model.

4.3 Hierarchical Consistency Performance

Table 3 further studies the consistency performance. Our method still maintains SOTA consistency performance in the absence of labeled training corpora. It is clear that HGCLR and BERT (Vanilla FT) which uses the direct fitting method only achieve 0 points in PMicro-F1 and PMacro-F1 under the 1-shot setting. Compared with the best-performing baseline HierVerb, on average, HierICRF-BERT outperforms 7.08% and 4.25% PMicro-F1 scores, and 6.26% and 3.92% PMacro-F1, and 4.38% and 1.79% CMicro-F1, and 9.3% and 4.38% CMacro-F1 scores on WOS, DBpedia, respectively. As for HPT and HierVerb, directly extra embedding injection to the pre-trained LM pays less attention to the hierarchy consistency. The results highlight that the proposed paradigm allows model to optimize from a path routing perspective, more consideration is given to the label dependency during the process of iteratively transiting between layers on the hierarchically repeated series to better deal with the hierarchical inconsistency problem. Surprisingly, when more training data are given, the performance gap between HierICRF and all other baselines gradually decreases as we may hypothesize. We conjecture that it is because although all baseline methods lack domain-hierarchy adaptation in the paradigm, data-hungry based methods such as GNN can still learn hierarchical consistency dependency through directly overfitting, thereby gradually improving consistency performance.

4.4 Ablation Experiments

To illustrate the effect of our proposed mechanisms, we conduct ablation studies on WOS, as shown in Table 4. When both ICRF and CHR are removed, HierICRF is equivalent to Vanilla SoftVerb. Removing ICRF results in significant performance degradation, with an average of 1.72% Micro-F1 and 2.38% Macro-F1 on BERT and 1.72% Micro-F1 and 2.37% Macro-F1 on T5, meaning that ICRF plays an important role in incorporating hierarchy dependencies. Furthermore, the performance decreases even more when both ICRF and CHR are removed, e.g., in the 4-shot case, Macro-F1 even drops

Methods	WOS	
	Micro-F1	Macro-F1
HierICRF	**87.12**	**81.65**
HierVerb	87.00	81.57
HPT	87.10	81.44
SoftVerb	86.80	81.23
HGCLR	87.08	81.11
HiMatch-BERT	86.70	81.06
BERT (Vanilla FT)	85.63	79.07

Table 5: Full-shot experiments using `BERT-base-uncased` on WOS dataest .

I_{chain}	WOS		DBpedia	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
5	**73.83**	**65.40**	**95.14**	**91.20**
3	72.86	65.01	95.05	91.11
1	72.16	64.55	94.74	90.51
0	72.01	64.29	94.60	90.37

Table 6: Effect of Hierarchy-aware Chain's Length. Here we conduct experiments under 4-shot settings.

K	Method	WOS	
		Micro-F1	Macro-F1
1	HierICRF-T5	**62.39**	**51.13**
	HierICRF-BERT	62.10	49.51
	HierVerb	61.29	47.70
	HPT	49.75	19.78
	HGCLR	20.10	0.50
	BERT (Vanilla FT)	10.78	0.25
2	HierICRF-T5	**68.40**	**60.05**
	HierICRF-BERT	68.14	58.81
	HierVerb	67.92	56.92
	HPT	60.09	35.44
	HGCLR	44.92	3.23
	BERT (Vanilla FT)	20.50	0.34
4	HierICRF-T5	74.94	**68.27**
	HierICRF-BERT	**75.22**	67.81
	HierVerb	73.88	64.80
	HPT	69.47	53.22
	HGCLR	68.12	52.92
	BERT (Vanilla FT)	67.44	51.66

Table 7: We further conduct experiments with the `T5-large` and `BERT-large-uncased` on WOS.

by 6.57% compared to the full method deployed on BERT. This firmly highlights the significance of combining chain of hierarchy-aware reasoning and hierarchical iterative CRF allows HierICRF to gradually inject label hierarchy information by feeding back the accumulated error transition of label nodes in the process of path routing step-by-step.

4.5 Benefit in a Full-Shot Setup

We conduct experiments using a full-shot setting and use the hyperparameter set directly from the few-shot settings. For the baseline models, we reproduce their experiments based on the settings in their original paper. While the main focus of our work is on the performance under few-shot settings, it is worth noting that HierICRF surprisingly outperforms all baselines, such as HGCLR, HPT, and HierVerb, in the full-shot setting. As shown in Table 5, our Micro-F1 is slightly higher than HGCLR, HPT, and HierVerb, while Macro-F1 significantly beats them by 0.28% on average. Besides, HierICRF significantly outperforms BERT (Vanilla FT), HiMatch, and SoftVerb.

4.6 Effect of Reasoning Chain's Length

To further study the effects of the hierarchy-aware reasoning chain, we conduct experiments on the iteration length of the reasoning chain. As we can see in Table 6, the performances on both WOS and DBpedia degrade substantially as I_{chain} decreases, with an average Micro-F1 of 1.82% and 0.54% on WOS and DBpedia. The results show that our method can correct more hierarchical inconsistencies between time steps and feedback to model optimization when performing hierarchical transitions in longer hierarchically repeated series.

4.7 Effect of Model Scales

To further study the ability of HierICRF to utilize the prior knowledge of the PLMs, we conduct experiments on `BERT-large-uncased` and `T5-large`. Table 7 demonstrates that HierICRF consistently outperforms all baseline models in all shot settings. We find that the gap is even significantly larger for HierICRF and all other baseline models compared to using `BERT-base-uncased`. For example, compared with HierVerb, HierICRF-BERT(`BERT-large-uncased`) achieves a 1.53% Macro-F1 and a 0.45% Micro-F1 scores increase under 1-shot setting. But the improvements of Macro-F1 and Micro-F1 are 1.81% and 0.81% under `BERT-base-uncased`, respectively. Surprisingly, when using `T5-large`, we find that its performance improvement relative to `T5-base` is greater than that of the improvement obtained by all all BERT-based methods (including HierICRF-BERT) from `BERT-base-uncased` to `BERT-large-uncased`, and even achieve SOTA under no more than 2-shot settings. The findings further underscore that HierICRF outperforms all baseline models in effectively leveraging the prior knowledge embedded within larger language models. This advantage becomes even more pronounced as the scale of the language model increases, highlighting the significant impact of HierICRF's ability to harness this expansive prior knowledge.

5 Conclusions

In this paper, we study the challenge of domain-hierarchy adaptation. We propose a novel framework named HierICRF which elegantly leverages the prior knowledge of PLMs from the perspective of the path routing performed at the language modeling process for better few-shot domain-hierarchy adaptation and can be flexibly applied in any transformer-based architecture. We perform few-shot settings on HTC tasks and extensive experiments show that our method achieves state-of-the-art performances on 2 popular HTC datasets while guaranteeing excellent consistency performance. Moreover, our method provides a perspective for hierarchy-based tasks to integrate into a unified instruction tuning paradigm for pretraining. For future work, we decide to extend HierICRF for effective non-tuning algorithms of LLM.

Acknowledgments

We thank the reviewers for their insightful comments. This work was supported by National Science Foundation of China (Grant Nos.62376057) and the Start-up Research Fund of Southeast University (RF1028623234).All opinions are of the authors and do not reflect the view of sponsors.

References

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, and Nick et al. Ryder. Language models are few-shot learners. In *NeurIPS*, 2020.

[Cevahir and Murakami, 2016] Ali Cevahir and Koji Murakami. Large-scale multi-class and hierarchical product categorization for an e-commerce giant. In *COLING*, 2016.

[Chen *et al.*, 2021] Haibin Chen, Qianli Ma, Zhenxi Lin, et al. Hierarchy-aware label semantics matching network for hierarchical text classification. In *ACL*, 2021.

[Chung *et al.*, 2022] Hyung Won Chung, Le Hou, Shayne Longpre, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[Forney, 1973] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[Gao *et al.*, 2021] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *ACL*, 2021.

[Irsan and Khodra, 2019] Ivana Clairine Irsan and Masayu Leylia Khodra. Hierarchical multi-label news article classification with distributed semantic model based features. *International Journal of Advances in Intelligent Informatics*, 5(1):40–47, 2019.

[Ji *et al.*, 2023] Ke Ji, Yixin Lian, Jingsheng Gao, et al. Hierarchical verbalizer for few-shot hierarchical text classification. In *ACL*, 2023.

[Kowsari *et al.*, 2017] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, et al. Hdltex: Hierarchical deep learning for text classification. In *ICMLA*, 2017.

[Kowsari *et al.*, 2019] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, et al. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[Li *et al.*, 2022] Guozheng Li, Xu Chen, Peng Wang, Jiafeng Xie, and Qiqing Luo. Fastre: Towards fast relation extraction with convolutional encoder and improved cascade binary tagging framework. In *IJCAI*, 2022.

[Li *et al.*, 2023] Guozheng Li, Peng Wang, and Wenjun Ke. Revisiting large language models as zero-shot relation extractors. In *Findings of EMNLP*, 2023.

[Li *et al.*, 2024] Guozheng Li, Wenjun Ke, Peng Wang, Zijie Xu, Ke Ji, Jiajun Liu, Ziyu Shang, and Qiqing Luo. Unlocking instructive in-context learning with tabular prompting for relational triple extraction. *arXiv preprint arXiv:2402.13741*, 2024.

[Mao *et al.*, 2019] Yuning Mao, Jingjing Tian, Jiawei Han, et al. Hierarchical text classification with reinforced label assignment. In *EMNLP*, 2019.

[Nadeau and Sekine, 2007] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[Petroni *et al.*, 2019] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, et al. Language models as knowledge bases? In *EMNLP*, 2019.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, et al. Improving language understanding by generative pre-training. 2018.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[Scao *et al.*, 2022] Teven Le Scao, Angela Fan, Christopher Akiki, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[Schick and Schütze, 2021] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*, 2021.

[Sinha *et al.*, 2018] Koustuv Sinha, Yue Dong, Jackie Chi Kit Cheung, et al. A hierarchical neural attention-based text classifier. In *EMNLP*, 2018.

[Sutton *et al.*, 2012] Charles Sutton, Andrew McCallum, et al. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373, 2012.

[Wang *et al.*, 2021] Xuepeng Wang, Li Zhao, Bing Liu, et al. Concept-based label embedding via dynamic routing for hierarchical text classification. In *ACL*, 2021.

[Wang *et al.*, 2022a] Zihan Wang, Peiyi Wang, Lianzhe Huang, et al. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In *ACL*, 2022.

[Wang *et al.*, 2022b] Zihan Wang, Peiyi Wang, Tianyu Liu, et al. Hpt: Hierarchy-aware prompt tuning for hierarchical text classification. In *EMNLP*, 2022.

[Wang *et al.*, 2023] Peng Wang, Tong Shao, Ke Ji, Guozheng Li, and Wenjun Ke. fmlre: A low-resource relation extraction model based on feature mapping similarity calculation. In *AAAI*, 2023.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837, 2022.

[Wu *et al.*, 2019] Jiawei Wu, Wenhan Xiong, and William Yang Wang. Learning to learn and predict:

A meta-learning approach for multi-label classification. In *EMNLP*, 2019.

[Xu and Barbosa, 2018] Peng Xu and Denilson Barbosa. Neural fine-grained entity type classification with hierarchy-aware loss. In *NAACL*, 2018.

[Yu *et al.*, 2022] Chao Yu, Yi Shen, and Yue Mao. Constrained sequence-to-tree generation for hierarchical text classification. In *SIGIR*, 2022.

[Zhang *et al.*, 2022] Xinyi Zhang, Jiahao Xu, Charlie Soh, and Lihui Chen. La-hcn: label-based attention for hierarchical multi-label text classification neural network. *Expert Systems with Applications*, 187:115922, 2022.

[Zhou *et al.*, 2020] Jie Zhou, Chunping Ma, Dingkun Long, et al. Hierarchy-aware global model for hierarchical text classification. In *ACL*, 2020.

[Zhu *et al.*, 2023] Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. Solving math word problems via cooperative reasoning induced language models. In *ACL*, 2023.