Retrieval-style In-context Learning for Few-shot Hierarchical Text Classification

Huiyao Chen[1,3]*, **Yu Zhao**[2]*, **Zulong Chen**[3], **Mengjia Wang**[3], **Liangyue Li**[3], **Meishan Zhang**[1]†, **Min Zhang**[1]

[1]Institute of Computing and Intelligence, Harbin Institute of Technology (Shenzhen), China
[2]College of Intelligence and Computing, Tianjin University, China
[3]Alibaba Group, China
chenhy1018@gmail.com, zhaoyucs@tju.edu.cn, chenzulong198867@163.com
mason.zms@gmail.com, zhangmin2021@hit.edu.cn

Abstract

Hierarchical text classification (HTC) is an important task with broad applications, and few-shot HTC has gained increasing interest recently. While in-context learning (ICL) with large language models (LLMs) has achieved significant success in few-shot learning, it is not as effective for HTC because of the expansive hierarchical label sets and extremely ambiguous labels. In this work, we introduce the first ICL-based framework with LLM for few-shot HTC. We exploit a retrieval database to identify relevant demonstrations, and an iterative policy to manage multi-layer hierarchical labels. Particularly, we equip the retrieval database with HTC label-aware representations for the input texts, which is achieved by continual training on a pretrained language model with masked language modeling (MLM), layer-wise classification (CLS, specifically for HTC), and a novel divergent contrastive learning (DCL, mainly for adjacent semantically similar labels) objective. Experimental results on three benchmark datasets demonstrate superior performance of our method, and we can achieve state-of-the-art results in few-shot HTC.

1 Introduction

Hierarchical text classification (HTC), a specialized branch of multi-label text classification, involves the systematic arrangement and categorization of textual data throughout a tiered label structure. The output labels are organized in a parent-child hierarchy, with the higher-level labels encompassing broader concepts, and the child labels delineating more specific subtopics or attributes. In recent years, HTC has gained significant attention, due to its applicability across a variety of fields, including recommendation systems (Sun et al., 2023; Agrawal et al., 2013), document categorization (Peng et al., 2016; Kowsari et al., 2017), and information retrieval (Sinha et al., 2018).

In standard supervised HTC, there is an underlying assumption of abundant training samples (Zhao et al., 2023; Im et al., 2023; Song et al., 2023), which is often unattainable and expensive to construct manually. Moreover, HTC datasets are characterized by a complex hierarchical label structure, with leaf labels typically following a Zipfian distribution, resulting in very few data instances for these labels. As a result, the few-shot setting is more realistic, and has gained increasing interest recently (Ji et al., 2023; Bhambhoria et al., 2023; Wang et al., 2023b). Nevertheless, existing works often struggle with unsatisfactory performance in this setting. For example, BERT with the vanilla fine-tuning strategy performs extremely poorly in few-shot HTC.

Recently, large language models (LLMs) have achieved notable success on various NLP tasks (Wang et al., 2023a; Drozdov et al., 2023; Zeng et al., 2023), which have significantly enhanced the efficacy of in-context learning (ICL) with relevant demonstrations in the few-shot setting (Shome and Yadav, 2023; Dai et al., 2023; Zhang et al., 2023). However, the application of ICL on HTC faces unique challenges, diverging from traditional text classification scenarios. These challenges are primarily due to two distinct characteristics of HTC, as delineated in Figure 1. Firstly, HTC features a deep hierarchical labeling structure and expansive label sets, resulting in large label sets in ICL, which adversely impacts its performance. Secondly, as the hierarchy deepens, the semantic similarity between adjacent labels increases (Stein et al., 2019), making it very

* Equal contribution
† The corresponding author.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 1214–1231, 2024. https://doi.org/10.1162/tacl_a_00697
Action Editor: Byron Wallace. Submission batch: 12/2023; Revision batch: 3/2024; Published 9/2024.

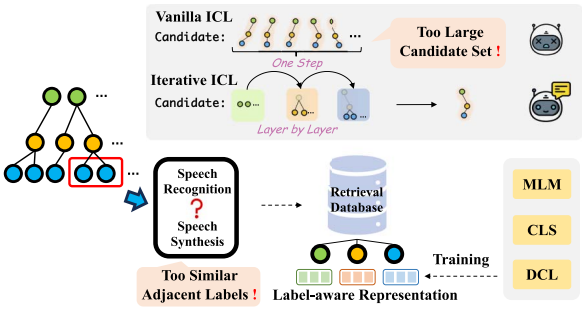

Figure 1: The problems of ICL-based few-shot HTC and our solutions. MLM, CLS, and DCL denote **M**ask **L**anguage **M**odeling, Layer-wise **CL**a**S**sification, and **D**ivergent **C**ontrastive **L**earning, which are the three objectives for indexer training.

challenging to select relevant demonstrations that guide the learning process efficiently.

In this work, we introduce the first ICL-based framework for few-shot HTC. Specifically, we use a LLM as the foundation model for inference, and provide demonstrations to guide HTC label generation through ICL. Our success depends on finding suitable demonstrations for a given input. In order to achieve this, we build a retrieval database that can find the most-relevant demonstrations for the input. Further, in order to avoid providing an enormous set of multi-layer contextual HTC labels all at once, as is required for ICL, we suggest an iterative policy to infer the labels layer-by-layer, reducing the number of candidate labels greatly.

The quality of our retrieval database is highly critical. The key idea is to obtain the HTC label-aware representations for input texts, which are then used for subsequent retrieving. Given an input, we define prompt templates for each-layer HTC, concatenating them with the raw input to form a new text. The hidden vectors of prompts are exploited as label-aware representations. We perform continual training to learn our representations using three types of objectives: the masked language model (MLM), the multi-label classification (MLC) for HTC particularly and a well-designed divergent contrastive learning (DCL) objective. The DCL is especially useful for the semantically closed HTC labels (e.g., adjacent labels from the same parent). In addition, we can incorporate label descriptions naturally by DCL, which can benefit ICL-based HTC much.

We conduct experiments on two classic English hierarchical classification datasets, WOS (Reuters, 2012) and DBpedia (Sinha et al., 2018), as well as a Chinese patent dataset,[1] and measure model performance by both micro-F1 and macro-F1 metrics consistent with previous work. Results show that our method is highly effective, giving improved results compared with a series of baselines for the few-shot HTC with different shots. We can also achieve state-of-the-art (SOTA) results on these HTC datasets. Further, we perform thorough qualitative analysis in order to gain a comprehensive understanding of our methodology.

In summary, our contributions can be summarized as follows:

- We present the first ICL-based work for HTC backended with LLMs, utilizing the knowledge from the existing demonstrations during the LLM inference.
- We propose a novel retrieve-style framework for ICL of HTC, which can help to find highly relevant demonstrations to support HTC label prediction.
- We conduct extensive experiments to compare our method with several representative baselines, and results show that our method can advance the SOTA of HTC further.

Our code is publicly available at `https://github.com/DreamH1gh/TACL2024` to facilitate future research.

2 Related Work

HTC is initially presented by Koller and Sahami (1997), and neural network models have achieved great advances in this task. Previous approaches treat HTC as multi-label classification and adopt traditional text classification methods for HTC problems (Aly et al., 2019; Zhang et al., 2022b). The majority of recent HTC studies pay the focus on finding ways to insert the hierarchical label knowledge into the model (Chen et al., 2021; Mao et al., 2019; Sinha et al., 2018). Several studies attempt to solve HTC problems by modeling the hierarchical labels as a graph or a tree structure (Zhou et al., 2020; Tai et al., 2015) while other researchers try to apply meta-learning (Wu et al., 2019) or reinforcement learning (Mao et al., 2019) to leverage HTC label structure.

[1]This dataset is not allowed to be publicly released due to local law.

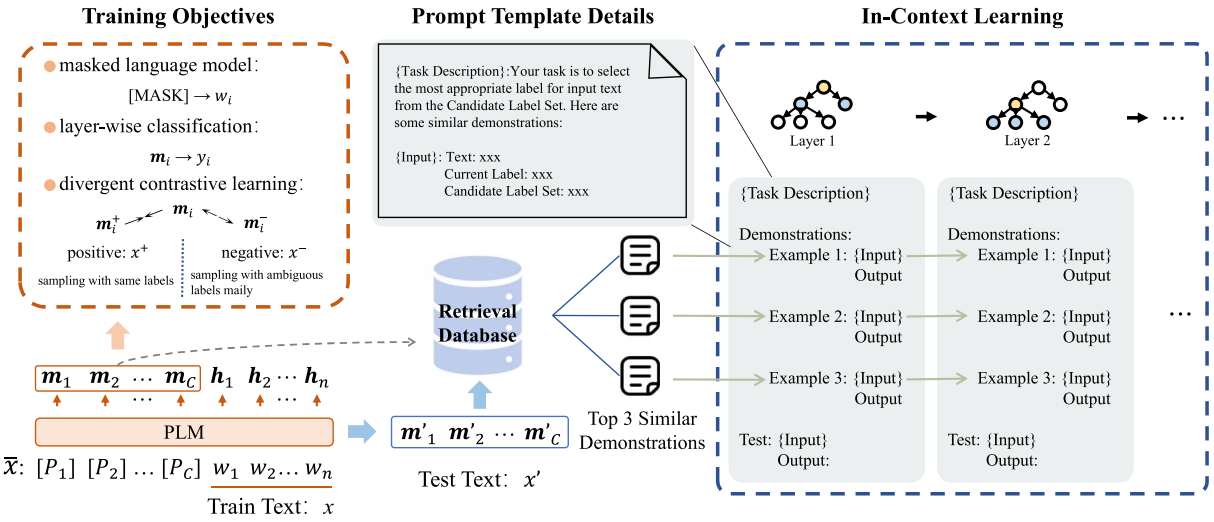

Figure 2: The architecture of retrieval-style in-context learning for HTC. The [P$_j$] term is a soft prompt template token to learn the j-th hierarchical layer label index representation.

However, existing methods mainly concentrate on encoding the holistic label structure, ignoring the classification of nuanced long-tail terminal labels. There have been efforts prove that retrieval augmented methods could help classification task with only few-shot samples (Chen et al., 2022; Zhang et al., 2022a; Yu et al., 2023; Xiong et al., 2023), which could be a solution of the long-tail challenge. Given this insight, we make an attempt to convert the HTC task to a retrieval form.

Moreover, with the development of LLMs, recent work explores solutions that tackle traditional NLP tasks with the ICL paradigm and achieve surprising effectiveness (Shome and Yadav, 2023; Fei et al., 2023; Min et al., 2022; Liu et al., 2022). But ICL strongly relies on demonstration selection (Gao et al., 2021; Zhao et al., 2021; Rubin et al., 2022; Zhang et al., 2022c; Li et al., 2023). Many studies explore adjusting demonstrations for better performance through instruction formatting (Zhou et al., 2023), example ordering (Liu et al., 2021), and demonstration filtering (Sorensen et al., 2022). In our work, we combine the ICL-based framework with retrieval for HTC, selecting demonstrations that involve both the language knowledge of LLM and advantages of retrieval.

3 Method

Problem Formulation In HTC tasks, the structure of labels $\mathcal{H} = (\mathcal{Y}, E)$ is often predefined as a tree, where $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_L\}$ is a set of nodes (labels) and E indicates the parent-child hierarchical edges (connections) between the labels. It is worth noting that in the label structure, every node, except for the root, has one and only one parent. Generally speaking, HTC tasks select the label path in $\mathcal{H}$ for a given text x. We define that $x = w_1 w_2 \cdots w_n$ is a text and $y = \{y_1, y_2, \ldots, y_C\} \subseteq \mathcal{Y}$ is the corresponding hierarchical labels which follow $\mathcal{H}$, where C denotes the maximum label depth.

Proposed Framework Figure 2 illustrates our ICL-based framework for HTC. We first train a PLM-based indexer and build a retrieval database containing reference samples (the training data). After that, we perform a similarity search in the retrieval database with the text to be inferred. Finally, we construct an ICL prompt with highly similar demonstrations for prediction.

We introduce our ICL prompt policy for HTC (§ 3.1), and then detail the retrieval database construction (§ 3.2) and demonstration retrieval methods (§ 3.3).

3.1 In-context Learning

In order to integrate label structural information into ICL, we propose an iterative approach by decoupling the label structure $\mathcal{H}$. We decompose the label structure into several subclusters, each corresponding to a parent-child set. Then, we employ an iterative method to produce the sub-labels layer by layer until we arrive at the leaf labels.

As shown in Figure 2, we perform iterative inference at each hierarchy level. Based on the

Top K similar demonstrations, the prompt contains K identical structured text blocks. Each block contains three parts: `Text`, `Current Label`, and `Candidate Label Set`. `Text` is the demonstration content. `Current Label` is the predicted label of the previous hierarchy level.[2]

When the LLM is used for inference in classification tasks, the entire set of labels is always presented. The inference result is drawn from this large label set. In contrast, our method supplies a pruned subset of labels as a concise candidate label set. `Candidate Label Set` is the intersection of the child nodes of the current label and the selected K demonstration labels, which maximizes the use of demonstration information and avoids the impact of erroneous labels. The predicted label of the next hierarchy level is required to be selected from the candidate label set.

3.2 Retrieval Database Construction

After determining the ICL prompt policy, it is crucial to obtain demonstrations related to the test text, which will provide effective guidance for LLM inferences. Firstly, we train a HTC indexer to generate index vectors for each training sample. We employ a pretrained text encoder as the indexer and use a prompt template to elicit multi-layer representations as index vectors. To make the index vectors discriminative, the indexer is trained via DCL based on label descriptions.

Index Vector Representation. To further utilize the language knowledge embedded in pre-trained text encoders and leverage interdependencies among hierarchical labels, we propose the construction of a concise prompt template prior to raw input x. The new text is formatted as: $\bar{x} =$ `[P1] [P2] ... [PC]` x. Here, the `[Pj]` term is a soft prompt template token to learn the j-th hierarchical layer label index representation. Then, we input $\bar{x}$ into the encoder of PLM to obtain the hidden states:

$$\boldsymbol{m}_1 \cdots \boldsymbol{m}_C \, \boldsymbol{h}_1 \cdots \boldsymbol{h}_n = \text{PLM}(\bar{x}). \quad (1)$$

Thus, we can obtain the index vectors $\boldsymbol{m}_1 \cdots \boldsymbol{m}_C$ consisting of hidden state embeddings for all fixed-position `[P]`, where we consider $\boldsymbol{m}_j$ as the index vector of the j-th hierarchical level corresponding to x.

[2] `Current Label` is Root when predicting the first hierarchy level.

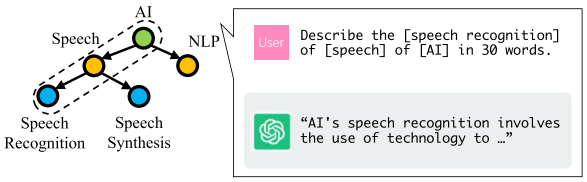

Figure 3: Label description generation.

Label Description. In order to reduce the ambiguity errors caused by insufficient label information, we explore diverse approaches that aim to provide more informative and representative label information for HTC task. First, we propagate the textual information of all label nodes to their corresponding leaf nodes, obtaining the textual information with the entire label path. As shown in Figure 3, for the original leaf label ''speech recognition'', its label path is ''speech recognition of speech of AI''.

However, due to the close semantic proximity of adjacent leaf node labels, the generated label path may still be insufficient or ambiguous. For example, ''speech recognition of speech of AI'' and ''speech synthesis of speech of AI'' may still be difficult to distinguish. To address this issue, we use the LLM to expand and enhance the label path l of x by leveraging the knowledge contained within the LLM:

$$d = \text{LLM}(\texttt{Describe}, l), \quad (2)$$

where d is the description of the label path l and `Describe` denotes the prompt used to generate the description. By utilizing expanded and enhanced label descriptions, we could obtain a more detailed explanation of the label.

Indexer Training. For indexer training, we apply the objectives of mask language modeling $\mathcal{L}_{mlm}$, and layer-wise classification $\mathcal{L}_{cls}$. $\mathcal{L}_{mlm}$ is used to predict the words that fill the random mask tokens in the inputs. $\mathcal{L}_{cls}$ is to predict HTC labels through each hierarchical layer index vectors.

Additionally, we propose DCL for indexer training, which uses label text information to select positive and negative samples. For x, positive samples are chosen from sentences with the same label as x. Additionally, the corresponding label description d could be treated as a positive sample. Negative samples consist of two parts. First, based on the similarity between d and descriptions of other labels, negative examples are

sampled from highly similar label categories. Similarly, their corresponding label descriptions could be also treated as negative samples. In addition, a few randomly selected sentences from other labels are used as negative samples of x. Thus, compared to traditional random sampling methods, our negative sample selection approach opts for more instances with semantically similar labels as hard negative samples.

Then the index vectors among the positive samples are pulled together and the negative ones are pushed apart. Taking x as an example, denote $\boldsymbol{B} = \{x, x^+, x_1^-, \ldots, x_n^-\}$ as a group of input data. The contrastive loss can be calculated as:

$$\mathcal{L}_{\text{con}} = -\sum_{j}^{C} \log \frac{e^{\cos(\boldsymbol{m}_j, \boldsymbol{m}_j^+)/\tau}}{\sum_k^n e^{\cos(\boldsymbol{m}_j, \boldsymbol{m}_{j,k}^-)/\tau}}, \quad (3)$$

where $cos(\cdot, \cdot)$ is the cosine similarity, τ is the contrastive learning temperature. In comparison to calculating the contrastive loss in a random sampling batch, our DCL pays more attention to samples whose labels are less similar to the x.

The final objective is set in the multi-task form:

$$\mathcal{L} = \mathcal{L}_{\text{mlm}} + \alpha\mathcal{L}_{\text{cls}} + \beta\mathcal{L}_{\text{con}}. \quad (4)$$

After the training step, we store index vectors $\boldsymbol{m_1} \cdots \boldsymbol{m_C}$ of each training instance to construct the retrieval database.

3.3 Demonstration Retrieval

With the database and indexer in hand, we can process predictions in retrieval form. For the test text x', we also use the trained indexer to obtain the hierarchical index vectors $\boldsymbol{m}'_1 \cdots \boldsymbol{m}'_C$. Then, we select similar instances from the retrieval database by calculating similarity between their index vectors. For each training instance x, we have C index vectors $\boldsymbol{m}_1 \cdots \boldsymbol{m}_C$ in retrieval database. The similarity between x and x' can be calculated as:

$$\text{sim}(x, x') = \sum_{j}^{C} \frac{2^{j-1}}{2^C - 1} \cdot \cos(\boldsymbol{m}_j, \boldsymbol{m}'_j), \quad (5)$$

where the first factor is utilized to adjust the weights of similarity between different hierarchical layer, while ensuring that $\sum_j^C \frac{2^{j-1}}{2^C-1} = 1$. As the hierarchy deepens, the impact of index vector similarity gradually increases. Then, we choose the Top k most similar instances from the database as demonstrations. It is worth noting that we filter out instances with the same label here, to ensure that the labels of the Top k instances are different, providing relatively diverse instances for ICL.

Statistics	WOS	DBpedia	Patent
#levels	2	3	4
#Number of documents	46,985	381,025	30,104
#Level 1 Categories	7	9	10
#Level 2 Categories	134	70	17
#Level 3 Categories	NA	219	105
#Level 4 Categories	NA	NA	305
#Mean label length	1.8	1.7	4.4
#Max label length	3	7	14
#Mean document length	200.7	806.9	335.1
#Max document length	1262	881	1669

Table 1: Overview of HTC datasets.

4 Experiments

4.1 Settings

Dataset and Evaluation Metrics. Our experiments are evaluated on three datasets: Web-of-Science (WOS) (Reuters, 2012), DBpedia (Sinha et al., 2018), and Patent. WOS and DBpedia are both widely used English datasets for HTC and Patent which we collected consists of 30,104 Chinese patent records. We evaluate the effectiveness of our proposed method on both English and Chinese datasets. All of them are for single-path HTC. The statistics are illustrated in Table 1. Following the previous work, we report experimental results with Micro-F1 and Macro-F1.

Model Details. We utilize bert-base-uncased[3] (Devlin et al., 2019) as the base indexer for WOS and DBpedia datasets, while for Patent dataset, we employ chinese-bert-wwm-ext[4] (Cui et al., 2019, 2020). Regarding LLM, we select vicuna-7b-v1.5-16k[5] (Zheng et al., 2023) and gpt-3.5-turbo-0613,[6] which performs well on English for WOS and DBpedia datasets, and ChatGLM-6B[7] (Zeng et al., 2022; Du et al., 2022),

[3] https://huggingface.co/bert-base-uncased.

[4] https://huggingface.co/hfl/chinese-bert-wwm-ext.

[5] https://github.com/lm-sys/FastChat.

[6] https://openai.com/blog/chatgpt. ICL inference in experiments is based on this model unless otherwise specified.

[7] https://github.com/THUDM/ChatGLM-6B.

Algorithm 1: Sampling for HTC Few-shot

Input: Shot number: Q, Complete HTC dataset: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$

Output: Q-shot sampling dataset: $\mathcal{S}$

```
// Categorize samples by label path
Label path dictionary: C = {};
for i = 1 to N do
    if y_i not key in C then
        C = C ∪ {y_i : {x_i}};
    else
        C[y_i] = C[y_i] ∪ {x_i};
    end
end
// Q-shot random sampling
Q-shot sampling dataset: S = {};
for i = 1 . . . until all keys in C are traversed do
    if Count(C[y_i]) ≤ Q then
        S = S ∪ C[y_i];
    else
        S = S ∪ Random Sample(C[y_i], Q);
    end
end
return S
```

the top-performing open-source Chinese language model for Patent (due to legal restrictions, we can only evaluate it on open-source models). Our model is implemented with the OpenPrompt toolkit (Ding et al., 2022).

Experimental Settings. As mentioned in the introduction, we try to validate the effectiveness of our proposed method on the few-shot classes in the long-tail phenomenon. Specifically, we focus on the few-shot setting, where only Q samples per label path are available for training and use the same seeds as Ji et al. (2023), as shown in Algorithm 1. We conduct experiments based on Q $\in \{1, 2, 4, 8, 16\}$. The batch size of our proposed method is 1. It is composed of a training sample, a randomly selected positive sample from the same label, 4 randomly selected negative samples from the Top4 labels based on label description similarity, and 10 randomly selected negative samples from other labels. For all datasets, the learning rate is $5 * 10^{-5}$ and we train the model for 20 epochs and apply the Adam optimizer (Kingma and Ba, 2015) with a linearly decaying schedule with warmup steps at 0. The temperature of Vicuna, GPT3.5 and ChatGLM are both 0.2. α in Equation 4 is 1 and β is 0.01.

Baselines. In this work, we select several recent models as baselines:

- **BERT** with vanilla fine-tuning transforms HTC into a multi-label classification task. It is a standard method for HTC.
- **HiMatch** (Chen et al., 2021) learns the representation of text and labels separately and then defines different optimization objectives based on them to improve HTC.
- **HGCLR** (Wang et al., 2022a) incorporates the hierarchical label structure directly into the text encoder and obtains the hierarchy-aware text representation for HTC.
- **HPT** (Wang et al., 2022b) leverages a dynamic virtual template with soft-prompt label words and a zero-bounded multi-label cross-entropy loss, ingeniously aligning the goals of HTC and MLM.
- **HierVerb** (Ji et al., 2023) treats HTC as a multi-label problem at different levels, utilizing vectors as constrained by the hierarchical structure, effectively integrating knowledge of hierarchical labels.
- **EPR** (Rubin et al., 2022) estimates the output probability based on the input and a candidate training example prompt, separating examples as positive and negative and allowing effective retrieval of training examples as prompts during testing.
- **REGEN** (Yu et al., 2023) employs a retrieval model and a classification model, utilizing class-specific verbalizers and a general unlabeled corpus to enhance semantic understanding. Notably, REGEN incorporates supplementary unsupervised data.[8]

Retrieval employs our retrieval method to select the label associated with the text in the retrieval database that has the highest similarity score as the label for the test text.

Retrieval-style ICL involves the selection of the top three (K = 3) documents with distinct labels from the retrieval database. Subsequently, these documents and labels are utilized as demonstrations to construct the prompt, and our iterative method is applied to hierarchical label inference.

It is worth mentioning that in our LLM generative approach, if the generated label is not present in the candidate label set, the label corresponding to the retrieval text with the highest similarity is selected as its inference result.

[8] We employ full dataset texts, excluding the training data, as unsupervised data.

Q	Method	WOS(Depth 2)		DBpedia(Depth 3)	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1
1	BERT †	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 †	43.44 ± 8.09 (48.26)	7.71 ± 4.90 (9.32)	–	–
	HGCLR †	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 †	50.05 ± 6.80 (50.96)	25.69 ± 3.31 (27.76)	72.52 ± 10.20 (73.47)	31.01 ± 2.61 (32.50)
	HierVerb †	58.95 ± 6.38 (61.76)	44.96 ± 4.86 (48.19)	91.81 ± 0.07 (91.95)	85.32 ± 0.04 (85.44)
	EPR	31.77 ± 3.15 (35.31)	6.61 ± 2.70 (9.66)	16.58 ± 8.94 (25.60)	7.41 ± 4.13 (11.91)
	REGEN	5.62 ± 2.98 (8.70)	2.59 ± 2.45 (4.71)	18.70 ± 8.19 (27.33)	8.17 ± 3.87 (12.20)
	Retrieval	63.46 ± 2.30 (65.99)	50.24 ± 2.21 (52.66)	93.68 ± 0.05 (93.74)	88.41 ± 0.23 (88.67)
	Retrieval-style ICL	**68.91 ± 0.48 (69.38)**	**57.41 ± 0.40 (57.82)**	**94.54 ± 0.03 (94.58)**	**89.75 ± 0.09 (94.83)**
2	BERT †	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 †	46.41 ± 1.31 (47.23)	18.97 ± 0.65 (21.06)	–	–
	HGCLR †	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 †	57.45 ± 1.89 (58.99)	35.97 ± 11.89 (39.94)	90.32 ± 0.64 (91.11)	81.12 ± 1.33 (82.42)
	HierVerb †	66.08 ± 4.19 (68.01)	54.04 ± 3.24 (56.69)	93.71 ± 0.01 (93.87)	88.96 ± 0.02 (89.02)
	EPR	36.04 ± 2.97 (39.11)	16.28 ± 1.94 (18.32)	21.89 ± 5.02 (27.02)	15.96 ± 2.96 (19.02)
	REGEN	49.55 ± 2.88 (52.64)	12.12 ± 3.54 (15.91)	87.91 ± 2.44 (90.57)	71.80 ± 2.41 (74.35)
	Retrieval	69.85 ± 0.63 (70.58)	58.64 ± 0.58 (59.25)	94.12 ± 0.18 (94.32)	89.33 ± 0.19 (89.54)
	Retrieval-style ICL	**71.68 ± 0.09 (71.76)**	**61.99 ± 0.10 (62.08)**	**94.87 ± 0.10 (94.97)**	**90.82 ± 0.08 (90.89)**
4	BERT †	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 †	57.43 ± 0.01 (57.43)	39.04 ± 0.01 (39.04)	–	–
	HGCLR †	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 †	65.57 ± 1.69 (67.06)	45.89 ± 9.78 (49.42)	94.34 ± 0.28 (94.83)	90.09 ± 0.87 (91.12)
	HierVerb †	72.58 ± 0.83 (73.64)	63.12 ± 1.48 (64.47)	94.75 ± 0.13 (95.13)	90.77 ± 0.33 (91.43)
	EPR	38.42 ± 0.91 (39.36)	19.94 ± 1.32 (21.31)	27.94 ± 1.47 (29.56)	18.31 ± 1.70 (20.09)
	REGEN	58.75 ± 2.04 (60.71)	33.20 ± 2.01 (35.40)	94.11 ± 0.79 (95.01)	86.76 ± 1.04 (87.92)
	Retrieval	75.37 ± 0.70 (76.08)	65.94 ± 0.57 (66.41)	95.15 ± 0.07 (95.23)	91.26 ± 0.14 (91.38)
	Retrieval-style ICL	**75.62 ± 0.15 (75.78)**	**66.34 ± 0.09 (66.41)**	**95.26 ± 0.07 (95.23)**	**91.42 ± 0.05 (91.47)**
8	BERT †	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 †	69.92 ± 0.01 (70.23)	57.47 ± 0.01 (57.78)	–	–
	HGCLR †	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 †	76.22 ± 0.99 (77.23)	67.20 ± 1.89 (68.63)	95.49 ± 0.01 (95.57)	92.35 ± 0.03 (92.52)
	HierVerb †	78.12 ± 0.55 (78.87)	69.98 ± 0.91 (71.04)	95.69 ± 0.01 (95.70)	92.44 ± 0.01 (92.51)
	EPR	41.35 ± 0.43 (41.83)	22.19 ± 0.32 (22.57)	44.95 ± 0.43 (45.42)	31.13 ± 0.38 (31.56)
	REGEN	67.91 ± 1.47 (69.54)	55.39 ± 1.86 (57.32)	95.24 ± 0.12 (95.38)	90.56 ± 0.39 (90.99)
	Retrieval	**79.04 ± 0.48 (79.53)**	**70.59 ± 0.52 (71.04)**	**95.71 ± 0.06 (95.78)**	**92.50 ± 0.02 (92.52)**
	Retrieval-style ICL	76.93 ± 0.05 (76.98)	67.54 ± 0.04 (67.57)	95.43 ± 0.01 (95.44)	91.85 ± 0.01 (91.86)
16	BERT †	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 †	77.67 ± 0.01 (78.24)	68.70 ± 0.01 (69.58)	–	–
	HGCLR †	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 †	79.85 ± 0.41 (80.58)	72.02 ± 1.40 (73.31)	96.13 ± 0.01 (96.21)	93.34 ± 0.02 (93.45)
	HierVerb †	80.93 ± 0.10 (81.26)	**73.80 ± 0.12 (74.19)**	96.17 ± 0.01 (96.21)	93.28 ± 0.06 (93.49)
	EPR	44.57 ± 0.09 (44.70)	24.50 ± 0.18 (24.74)	52.68 ± 0.04 (52.71)	42.76 ± 0.03 (42.78)
	REGEN	77.64 ± 1.04 (78.70)	69.91 ± 1.68 (71.68)	95.88 ± 0.03 (95.91)	91.73 ± 0.07 (91.80)
	Retrieval	**81.12 ± 0.26 (81.38)**	73.72 ± 0.17 (73.82)	**96.22 ± 0.04 (96.27)**	**93.37 ± 0.02 (93.46)**
	Retrieval-style ICL	78.62 ± 0.03 (78.65)	69.56 ± 0.03 (69.59)	95.56 ± 0.00 (95.56)	92.04 ± 0.00 (92.04)

Table 2: Micro-F1 and Macro-F1 scores on two English datasets. We reported the average, standard deviation, and best results across three random seeds. **Bold**: the best result. Underlined: the second highest. †: the direct utilization of results from Ji et al. (2023).

4.2 Main Results

The main results are shown in Table 2 and Table 3. It can be observed that our retrieval-based approach achieved the best results across almost all settings. Also, we find that our method is less affected by random seeds, resulting in more stable and robust performance, which further demonstrated the effectiveness of our approach.

Specifically, as Q increases, all methods improve continuously. However, our retrieval-based method consistently performs the best, and its advantages become even more pronounced in extremely low-resource settings. In the 1-shot setting, compared with the previous state-of-the-art model, `Retrieval` shows an average of 4.51% micro, 5.28% macro-F1 absolute improvement on WOS, 1.87% micro, 3.09% macro-F1 absolute on

Q	Method	Patent(Depth 4) Micro-F1	Patent(Depth 4) Macro-F1
1	BERT	27.04 ± 1.48 (28.37)	2.40 ± 0.23 (2.67)
	HGCLR	28.99 ± 1.12 (29.89)	2.94 ± 0.24 (3.13)
	HPT	35.22 ± 1.07 (36.26)	4.22 ± 0.68 (4.87)
	HierVerb	41.83 ± 0.60 (42.52)	5.91 ± 0.65 (6.53)
	Retrieval	47.71 ± 0.41 (48.15)	10.76 ± 0.25 (11.02)
	Retrieval-style ICL	**52.23 ± 0.23 (52.45)**	**15.62 ± 0.17 (15.78)**
2	BERT	36.07 ± 0.24 (36.88)	6.41 ± 0.77 (6.92)
	HGCLR	36.73 ± 1.13 (38.03)	6.82 ± 0.21 (7.06)
	HPT	42.61 ± 0.75 (43.43)	10.53 ± 0.30 (10.87)
	HierVerb	48.42 ± 0.39 (48.74)	12.97 ± 0.39 (13.24)
	Retrieval	51.63 ± 0.34 (51.91)	15.12 ± 0.23 (15.32)
	Retrieval-style ICL	**56.84 ± 0.11 (56.93)**	**20.07 ± 0.10 (20.17)**
4	BERT	49.41 ± 0.98 (50.24)	9.64 ± 0.56 (10.13)
	HGCLR	50.24 ± 0.36 (50.63)	11.40 ± 0.29 (11.67)
	HPT	53.91 ± 0.44 (54.29)	18.45 ± 0.40 (18.79)
	HierVerb	57.58 ± 0.83 (58.64)	23.28 ± 0.39 (23.63)
	Retrieval	**60.53 ± 0.36 (60.82)**	**25.65 ± 0.29 (25.95)**
	Retrieval-style ICL	59.35 ± 0.10 (59.45)	24.15 ± 0.08 (24.25)
8	BERT	62.10 ± 1.34 (63.29)	26.85 ± 0.97 (27.75)
	HGCLR	64.69 ± 0.30 (65.01)	27.69 ± 0.47 (28.21)
	HPT	67.35 ± 0.13 (67.45)	28.39 ± 0.08 (28.46)
	HierVerb	68.74 ± 0.12 (68.82)	29.93 ± 0.07 (30.01)
	Retrieval	**69.44 ± 0.10 (69.53)**	**30.32 ± 0.07 (30.38)**
	Retrieval-style ICL	65.81 ± 0.05 (65.86)	27.86 ± 0.04 (27.89)
16	BERT	70.97 ± 0.36 (71.32)	30.90 ± 0.39 (31.34)
	HGCLR	71.44 ± 0.38 (71.74)	31.87 ± 0.05 (31.85)
	HPT	73.23 ± 0.17 (73.37)	33.44 ± 0.17 (33.60)
	HierVerb	75.72 ± 0.11 (75.85)	34.75 ± 0.10 (34.86)
	Retrieval	**75.94 ± 0.11 (76.06)**	**34.95 ± 0.05 (35.00)**
	Retrieval-style ICL	68.73 ± 0.03 (68.76)	29.82 ± 0.03 (29.85)

Table 3: Micro-F1 and Macro-F1 scores on the Chinese Patent dataset. We reported the average, standard deviation, and best results across three random seeds. **Bold**: the best result. Underlined: the second highest.

DBPedia, and 5.88% micro-F1, 4.85% macro-F1 absolute improvement on Patent. As the hierarchy depth increases, we find that /texttt[Retrieval] exhibits an advantage even in the 16-shot setting. We think it is because label text descriptions better differentiate categories, especially in the deeper HTC dataset. Despite the fact that our method is still the most effective, we observe that all methods in Patent don't perform well due to the deep hierarchy and the large number of labels. Furthermore, by examining the results on the Patent dataset, we observe that all methods exhibit similar trends to those on the English dataset, which also confirms the effectiveness of hierarchical classification approaches for Chinese HTC tasks. In the 1-shot, 2-shot, and 4-shot settings, `Retrieval-style ICL` achieves outstanding performance.

The EPR also uses a retrieval strategy. Following Rubin et al. (2022), we replicate its model on the HTC task[9], and it demonstrated inferior performance compared to our method. There are probably two factors causing this performance gap. On the one hand, EPR uses the poorly performing GPT-neo 2.7B as the scoring and inference LM and does not fine-tune it during the training process. Especially in the few shot setting, the ability of scoring LM itself has a significant impact on the experimental results. On the other hand, EPR is not proposed for HTC. Therefore, it does not utilize the information of hierarchical relationship between labels, which leads to a mismatch between the retrieved samples and the target sample. Furthermore, based on our observations, we find that the performance of EPR on the simpler dataset DBpedia is even inferior to that on WOS when compared to other methods. This discrepancy could be attributed to the deeper hierarchy of DBpedia, which leads to a larger number of labels and increases retrieval difficulty. In contrast, our proposed method incorporates classification objective loss and leverages hierarchical label information, which remains unaffected by these challenges and ensures more robust performance.

Observing Table 2, it can be observed that REGEN results, particularly in terms of Macro-F1, exhibit performance gaps compared to other HTC methods. This discrepancy stems from the fact that REGEN does not utilize label hierarchy information and focuses only on predicting leaf nodes. It reflects the significance of considering the label hierarchy structure of HTC.

4.3 Analysis

The Impact of Label Descriptions on Retrieval. We conduct experiments on different types of label texts: (1) original leaf label text, (2) all text on the label path, and (3) label descriptions generated by LLM. The results are shown in Figure 4.

We find that on WOS, (1)>(2)>(3), while on DBpedia and Patent datasets, (1)<(2)<(3). We analyze that it may be due to the shallow hierarchy and small number of labels in the WOS dataset. The label text itself has a high degree of discrimination, so adding additional information leads to a decrease. In contrast, the deeper hierarchy and larger number of labels in the DBpedia and Patent datasets require more information to distinguish the semantic meaning of label text. The experiment proves that label text improves retrieval results. However, which type of label

[9]The scoring LM utilized by EPR is GPT-neo, which does not perform well on Chinese. Therefore, we only present experimental results conducted on English datasets.

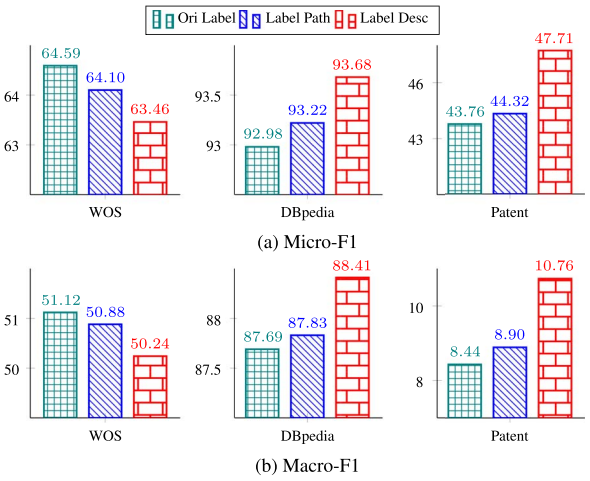

Figure 4: Results of different label text types in the 1-shot setting. `Ori Label` means the original leaf label text, `Label Path` means all text on the label path, and `Label Desc` means the label description text of LLM.

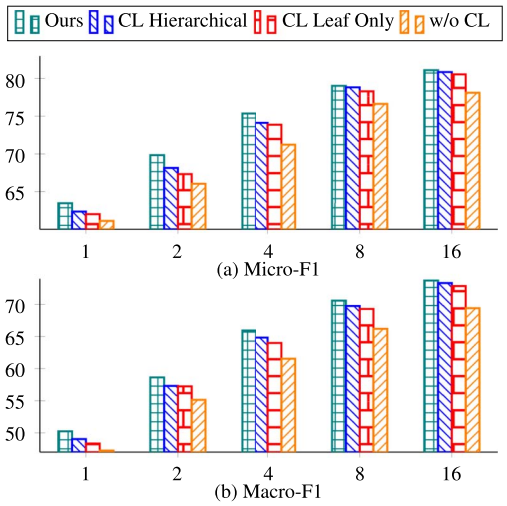

Figure 5: Results of different contrastive learning strategy on WOS dataset. The x-axis denotes the shot number Q and the y-axis denotes the F1 score.

text to use needs to be selected according to the dataset.

Comparison with Different Contrastive Learning Strategies. To demonstrate the effectiveness of our divergent contrastive learning, we illustrate the results with three more straightforward losses. `CL Hierarchical` denotes we calculate $\mathcal{L}_{con}$ for each hierarchical label representation with random sampling among a batch. `CL Leaf Only` refers we only calculate loss between leaf label representations with random sampling among a batch. `w/o CL` means training without $\mathcal{L}_{con}$ in Equation 4.

As shown in Figure 5, our divergent contrastive learning outperforms the others through all the shot numbers. Previous research has shown that contrastive learning is an effective option for training dense retrievers (Jin et al., 2023; Xiong et al., 2021). `w/o CL` has the lowest performance compared to other contrastive learning methods. As opposed to `CL Leaf Only` that treats HTC as a flat classification, `CL Hierarchical` models the label path information. Our divergent contrast learning selects more hard negative samples based on label similarity, further spatially pulling apart the vector distribution of samples with similar labels.

Comparison Between Classification-based and Retrieval-based Methods. Previous research on HTC has mainly used classification-based methods, which train classifiers to predict the probability distribution of each label. In contrast, our proposed retrieval-based method predicts labels by calculating similarity with a retrieval database to obtain the most similar text and corresponding labels. Therefore, we replaced our retrieval prediction with classifier prediction while keeping other settings consistent, and compared classification-based and retrieval-based methods. The results are shown in Table 4.

Q		Classification	Retrieval
1	Micro-F1	63.25 ± 2.17 (65.61)	**63.46 ± 2.30 (65.99)**
	Macro-F1	49.91 ± 2.43 (52.65)	**50.24 ± 2.21 (52.66)**
2	Micro-F1	69.09 ± 0.57 (69.74)	**69.85 ± 0.63 (70.58)**
	Macro-F1	58.49 ± 0.46 (59.04)	**58.64 ± 0.58 (59.25)**
4	Micro-F1	74.48 ± 0.74 (75.34)	**75.37 ± 0.70 (76.08)**
	Macro-F1	65.78 ± 0.60 (66.36)	**65.94 ± 0.57 (66.41)**
8	Micro-F1	78.36 ± 0.15 (78.48)	**79.04 ± 0.48 (79.53)**
	Macro-F1	70.55 ± 0.34 (70.93)	**70.59 ± 0.52 (71.04)**
16	Micro-F1	80.92 ± 0.21 (81.06)	**81.12 ± 0.26 (81.38)**
	Macro-F1	**73.88 ± 0.21 (74.08)**	73.72 ± 0.17 (73.82)

Table 4: The results of classification-based and retrieval-based methods on WOS dataset. We reported the average, standard deviation, and best results across three random seeds. **Bold**: the best result.

We find that under the few-shot setting, the retrieval-based method outperformed the classification-based method, although the gap gradually decreased with an increasing number of training samples. We speculate that in settings with a small number of samples, the classifier may not be well trained, while index vectors generated during the retrieval process have better semantic

Method	Level	DBpedia Q = 1 Micro-F1	DBpedia Q = 1 Macro-F1	DBpedia Q = 16 Micro-F1	DBpedia Q = 16 Macro-F1
BERT	1	17.60	5.13	98.42	94.55
	2	14.02	0.31	93.81	90.69
	3	11.17	0.10	90.13	85.92
EPR	1	23.51	11.51	71.40	69.51
	2	8.59	8.38	53.71	48.74
	3	7.06	6.51	43.67	39.71
Retrieval	1	98.34	95.08	98.75	96.58
	2	93.05	89.85	94.93	91.17
	3	89.14	87.62	90.66	87.01

Table 5: Results at different hierarchy levels on DBpedia dataset.

representations due to the rich semantic knowledge of pre-trained models. The retrieval-based method that utilizes similarity matching can achieve relatively better performance, especially in terms of Micro-F1.

Concurrently, we compare the classic classification method BERT, the retrieval-based method EPR, and our method on DBpedia, which has a deeper hierarchy. Table 5 illustrates the performance of these three methods at different hierarchical levels. It is observed that BERT performs well with slightly more samples, EPR excels with extremely limited samples, and our method consistently demonstrates excellent performance across scenarios.

Impact of Imbalanced Few-shot Sampling We sample few-shot training set with Algorithm 1, where we enforce balanced control over each type of samples. We now explore the impact of a bias training set and replace the step 13 to 17 in Algorithm 1 to:

$$\text{SampleN} = \text{Random}(0, \text{Min}(\mathcal{C}[y_i], Q))$$
$$\mathcal{S} = \mathcal{S} \cup \text{Random Sample}(\mathcal{C}[y_i], \text{SampleN}).$$

The report the results on WOS dataset in Table 6. We observe that the average F1 values of all the methods decrease and the random sampling approach makes a wider range of results. Our method could keep in lead and delivers a more stable performance.

Comparison with Zero-shot Setting on LLM. The quality of instances in the ICL prompt directly affects the results of ICL inference. We compared the impact of retrieval on ICL inference under different settings, and the results are

Q		WOS Micro-F1	WOS Macro-F1
16	BERT	70.42 ± 3.43 (74.07)	57.38 ± 7.98 (65.36)
	HiMatch	72.67 ± 4.97 (77.64)	61.80 ± 7.89 (69.80)
	HGCLR	71.93 ± 4.48 (76.41)	60.72 ± 5.83 (67.63)
	HPT	73.85 ± 4.33 (78.18)	65.02 ± 6.70 (73.39)
	HierVerb	75.63 ± 3.80 (79.62)	64.77 ± 7.60 (73.39)
	EPR	39.66 ± 5.17 (44.90)	15.67 ± 7.10 (23.13)
	Retrieval	**79.42** ± **3.82** (**83.81**)	**69.02** ± **5.56** (**74.92**)

Table 6: Results of randomly sampled 16-shot setting on WOS.

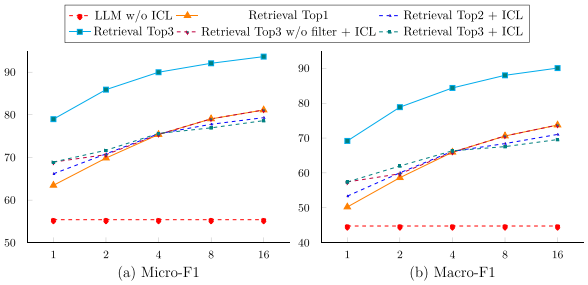

Figure 6: Micro-F1 (a) and Macro-F1 (b) results curves of top k retrieval and ICL with different numbers of examples on WOS. The horizontal axis represents the number of shots in the training set, and the vertical axis represents the metric value (%). `w/o` means 'without'.

shown in Figure 6. We distinguish between different methods using different lines, where the soild line represents the retrieval-based methods and the dashed line represents the LLM-based methods.

`LLM w/o ICL` refers to the situation where no examples are provided, and only the test document and the label set are given to the large model for inference. In other words, under the zero-shot setting, the inference relies entirely on the strong ability of LLM. Due to the large and complex label space, it is difficult to input it to the large model for inference at once. Therefore, `LLM w/o ICL` also uses iterative inference, sequentially inputting the label corresponding sub-clusters. We find that even under zero-shot setting, the large model still demonstrates strong performance, with 55.40% Micro-F1 and 44.79% Macro-F1, which even outperforms classification results of vanilla fine-tuned BERT under 4-shot.

Comparison with Different LLM Base Models. We also apply a powerful open access LLM base model Llama-7B for comparison. The results are shown in Table 7. `Llama-7B (Seq2Seq FT)` means we fine-tune the pre-trained Llama on our few-shot training set to generate hierarchical labels

Q	Model	Micro-F1	Macro-F1
1	BERT (Vanilla FT)	2.99 ± 20.85 (5.12)	0.16 ± 0.10 (0.24)
	Llama-7B (Seq2Seq FT)	42.76 ± 1.30 (44.10)	31.89 ± 1.24 (33.20)
	Retrieval (Top1)	63.46 ± 2.30 (65.99)	50.24 ± 2.21 (52.66)
	Llama-7B (Top3) ICL	65.61 ± 0.17 (65.80)	36.50 ± 0.48 (37.01)
	ChatGPT (Top3) ICL	**68.91 ± 0.48 (69.38)**	**57.41 ± 0.40 (57.82)**
2	BERT (Vanilla FT)	46.31 ± 0.65 (46.85)	5.11 ± 1.31 (5.51)
	Llama-7B (Seq2Seq FT)	45.66 ± 0.10 (45.77)	41.26 ± 0.11 (41.39)
	Retrieval (Top1)	69.85 ± 0.63 (70.58)	58.64 ± 0.58 (59.25)
	Llama-7B (Top3) ICL	67.09 ± 0.19 (67.29)	54.70 ± 0.30 (55.03)
	ChatGPT (Top3) ICL	**71.68 ± 0.09 (71.76)**	**61.99 ± 0.10 (62.08)**
4	BERT (Vanilla FT)	56.00 ± 4.25 (57.18)	31.04 ± 16.65(33.77)
	Llama-7B (Seq2Seq FT)	59.02 ± 0.08 (59.10)	52.63 ± 0.07 (52.66)
	Retrieval (Top1)	75.37 ± 0.70 (76.08)	65.94 ± 0.57 (66.41)
	Llama-7B (Top3) ICL	71.68 ± 0.09 (71.77)	60.22 ± 0.04 (60.26)
	ChatGPT (Top3) ICL	**75.62 ± 0.15 (75.78)**	**66.34 ± 0.09 (66.41)**
8	BERT (Vanilla FT)	66.24 ± 1.96 (67.53)	50.21 ± 5.05 (52.60)
	Llama-7B (Seq2Seq FT)	69.78 ± 0.05 (69.83)	63.22 ± 0.04 (63.26)
	Retrieval (Top1)	**79.04 ± 0.48 (79.53)**	**70.59 ± 0.52 (71.04)**
	Llama-7B (Top3) ICL	75.03 ± 0.04 (75.07)	63.58 ± 0.03 (63.61)
	ChatGPT (Top3) ICL	76.93 ± 0.05 (76.98)	67.54 ± 0.04 (67.57)
16	BERT (Vanilla FT)	75.52 ± 0.32 (76.07)	65.85 ± 1.28 (66.96)
	Llama-7B (Seq2Seq FT)	78.42 ± 0.19 (78.66)	70.09 ± 0.06 (70.15)
	Retrieval (Top1)	**81.12 ± 0.26 (81.38)**	**73.72 ± 0.17 (73.82)**
	Llama-7B (Top3) ICL	76.43 ± 0.05 (76.48)	65.56 ± 0.04 (65.60)
	ChatGPT (Top3) ICL	8.62 ± 0.03 (78.65)	69.56 ± 0.03 (69.59)

Table 7: The Micro-F1 and the Macro-F1 scores of the Llama model on WOS dataset. We reported the average, standard deviation, and best results across three random seeds. **Bold**: best result.

with a sequence-to-sequence target. `Llama-7B (Top3) ICL` means we use the ICL with our retrieved top 3 demonstrations on the fixed Llama model without fine-tune. The intricate architecture and extensive parameters of Llama-7B contribute to its superior performance over BERT (110M) in fine-tuning scenarios. In contrast, our retrieval model, built upon the BERT architecture with 110M parameters, consistently outperforms the fine-tuned results of Llama-7B.

In extremely few shot settings (such as Q= $1, 2, 4$), applying ICL on LLM with our retrieved results leads to further performance improvement, with more powerful models like ChatGPT typically demonstrating superior results. When Q grows, our retrieval methods could outperform LLM-based ICL. `Llama-7B (Top3) ICL` shows only marginal improvement compared to the `Retrieval (Top1)` result in 1-shot setting, implying that the degree of enhancement in ICL inference results is contingent upon the performance strength of the LLM.

The Improvement Limitation of ICL Inference. We present the Top1 and Top3 retrieval results[10] in Figure 6. For the retrieval-style ICL method, if we only provide the Top1 example retrieved, the ICL inference result will be consistent with Top1 retrieval result. Therefore, we present the results of `Retrieval Top2 + ICL` and `Retrieval Top3 + ICL` in Figure 6.

We show the results of constructing the candidate set without employing the filtering strategy, labeled as `Retrieval Top3 w/o filter + ICL`. When $Q = 1$, the Top3 retrieved labels are unique, rendering the results identical to `Retrieval Top3 + ICL`. When $Q = 2$, the Top3 retrieved labels typically encompass only two categories, leading to a scenario where one label (often the Top1 label) appears twice in the demonstration selections, introducing a bias in the inference process. As a result, `Retrieval Top3 w/o filter + ICL` slightly underperforms compared to `Retrieval Top2 + ICL`. When $Q \geq 4$, the Top3 retrieved labels usually belong to a single category, aligning the outcomes with those of `Retrieval Top1`.

Ideally, the ICL method can select the label closest to the gold-standard label from the candidate label set based on the provided examples. Taking Top3 as an example, the `Retrieval Top3 + ICL` curve should be close to the `Retrieval Top3` curve. In fact, the result curves of ICL are all round `Retrieval Top1`, and the curve of `Retrieval Top2 + ICL` is closer to `Retrieval Top1` than `Retrieval Top3 + ICL`. We analyze several possible reasons as follows: (1) Firstly, the LLM is not fine-tuned, and its understanding of labels may be inconsistent with the training set. (2) The effect of ICL is limited. The quality of retrieval examples is getting strong, resulting in increasingly similar candidate labels provided, which may increase the difficulty of the LLM inference. Figure 7 shows a case that LLM fails to choose the correct label, although the retrieved Top1 result is right. (3) Although LLM has demonstrated strong ability, there is still room for improvement. Using a more powerful LLM may yield better results. It can be concluded that our retrieval-style ICL method is far superior to direct inference using LLM, and can improve the performance on retrieval-based inference under extremely low resources. However, enhancing the retrieval results cannot continuously improve the performance of ICL.

The Impact of Different Prompts on LLM Inference. The differences in prompts directly affect the results of LLM inference. We conduct ablation experiments on the prompts we proposed

[10]Top3 selects the label with the highest overlap with the gold-standard label among the top3 retrieved labels as the predicted label result.

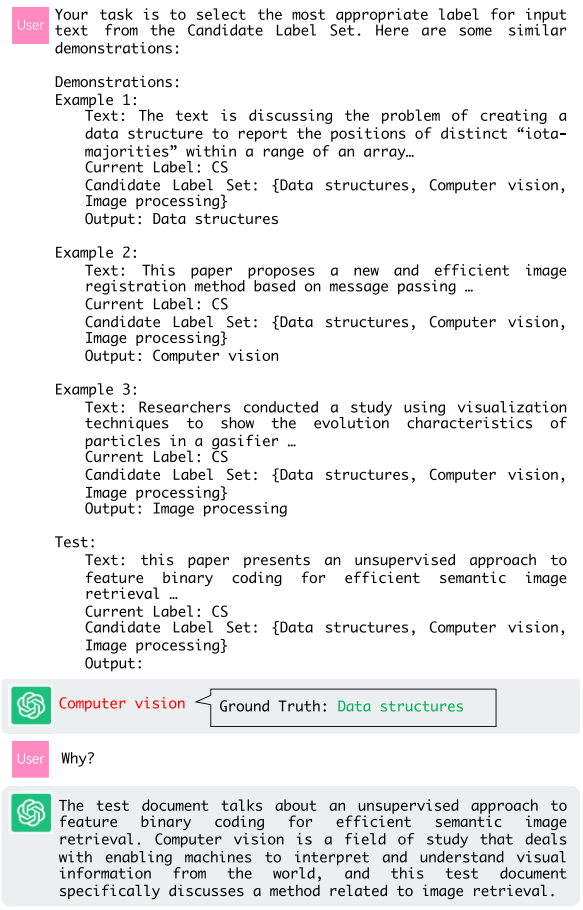

Figure 7: A case that LLM fails to choose the correct label although the retrieved Top1 result is right.

Q		WOS	
		Micro-F1	Macro-F1
0	LLM + iterative	**55.40**	**44.79**
	-w/o iterative	26.70	16.44
1	Random Samples + ICL	56.42	45.24
	Retrieval (Top3) + ICL	**68.91**	**57.41**
	-w/o iterative	68.52	57.06
	-w/o similar samples	64.75	52.27
	-w/o pruning	60.37	47.43
	-w/o candidate label set	52.35	32.66

Table 8: Results of different prompt settings on WOS. w/o means 'without'. **Bold**: best result.

to verify the rationality of our iterative prompts, and the results are shown in Table 8.

Under the zero-shot setting, when compared to directly inputting all hierarchical label paths to the language model, the iterative method improves the Micro-F1 by 28.70% and the Macro-F1 by 28.35%. It helps alleviate the negative impact of excessively long prompts during the inference process. This indicates that the iterative method is particularly effective for handling HTC tasks.

Taking 1-shot as an example, we randomly select three samples in the training dataset to form the prompt and use all labels from the target hierarchy layer as the candidate label set for iterative prediction. Interestingly, even with prompts constructed from random samples, the results obtained through ICL outperform LLM + iterative. This finding emphasizes the effectiveness of the ICL approach in generating inference results that closely match the desired format.

Then, we use our Top3 retrieval reulst as demonstrations and conduct four ablation comparisons. The first one is to remove the iterative operation, which means that the candidate label set consists of label paths, and all hierarchical labels are predicted at once. The second one is to remove all similar samples, and only provide the current label and candidate label set of the test document. The third one is to remove the pruning operation, which means that the candidate label set consists of all child labels of the current label. The last one is to remove the candidate label set and let the LLM select the most similar text, and use the label of the similar test as the label of the test document. The results prove that our prompts is reasonable, and each part of the prompt has a positive effect on inference.[11]

Retrieval-assisted Human Annotation. Furthermore, we recruit non-experts to annotate a portion of the test dataset, aiming to ascertain the expected upper-bound performance. We conduct experimental analyses on the WOS dataset as examples. We recruit college students with proficient English and conduct a simple annotation test, selecting nine annotators with comparable levels of annotation skill and efficiency, who are then divided into groups of three for subsequent annotation tasks. We randomly select 200 instances from the WOS test dataset for annotation. Three annotation methods are employed: (1) providing only the full list of labels; (2) based on (1), supplying an explanation for each label; (3) based on (2), offering the Top3 similar examples assisted by a retriever model trained on the 16-shot setting for annotation. Each annotation method is carried out by three annotators, with the final annotation results produced by voting.

[11] All details of prompts will be publicly available, thus enhancing the reproducibility of our work.

Annotation Methods	Micro-F1	Avg. Time (s)
(1) direct classification	67.25	37.2
(2) with label description	73.00	45.8
(3) 16-shot retrieval-assisted	86.44	9.1

Table 9: Statistical results of different annotation methods on WOS. Avg. Time indicates the average time (s) spent annotating each instance.

The statistical results of different annotation methods are presented in Table 9. Annotation method (1) represents the upper-bound result based on human knowledge under the 0-shot setting. A comparison reveals that after providing label descriptions, the Micro-F1 increases by 5.75%, but the average annotation time also lengthens due to the provision of more information. When assisted by the Top3 examples provided by a retriever trained in the 16-shot setting, the Micro-F1 significantly improves by 13.44%, and the average annotation time is reduced to only one-fifth of that for method (2), as many clearly incorrect labels are eliminated, reducing the difficulty of annotation. This indicates that our retrieval method can assist human annotation, effectively improving the quality of human annotations and reducing the time required.

Visualization of Index Vector. Finally, we use T-SNE (Van der Maaten and Hinton, 2008) to visualize the changes in `[P]` of the WOS test dataset before and after training, as shown in Figure 8. We find that index vectors exhibit clear hierarchical clustering characteristics, further demonstrating the effectiveness of our method.

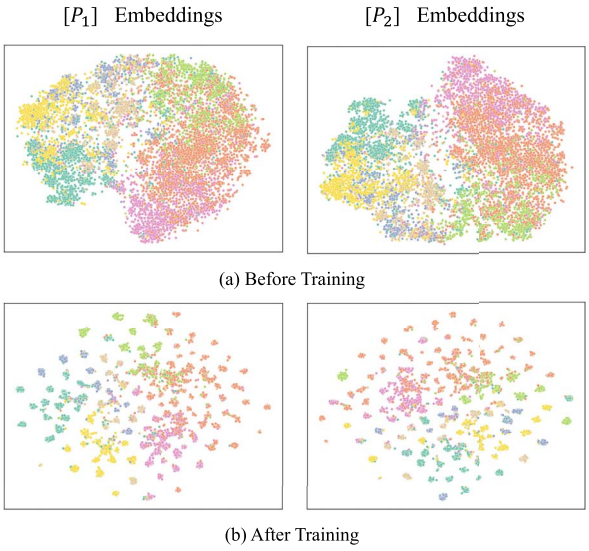

Figure 8: Visualization on the WOS test dataset. The top two figures show `[P]` embeddings obtained using the original BERT, while the bottom two figures show `[P]` embeddings obtain after training by our method.

5 Conclusion

In this paper, we proposed a retrieval-style ICL framework for few-shot HTC. We uniquely identify the most relevant demonstrations from a retrieval database to improve ICL performance and meanwhile designed an iterative policy to inference hierarchical labels sequentially, significantly reducing the number of candidate labels. The retrieval database is achieved by using a HTC label-aware representation for any given input, enabling the differentiation of semantically closed labels (especially the leaf adjacent labels). The representation learning is implemented by continual training on a PLM with three carefully designed objectives including MLM, layer-wise classification, and a novel DCL objective.

We conducted experiments on three benchmark datasets to evaluate our method. The results show that our method is highly effective, which is able to gain large improvements among a serious of baselines. Finally, our method can bring the state-of-the-art results in few-shot HTC on the three datasets. Further, we performed comprehensive analysis for deep understanding of our method, spreading various important factors.

This work still includes several unresolved problems, which might be addressed in the future. First, LLMs are currently confined to expanding text via label descriptions and their application to full training set expansion has not been effective. In order to fully utilize LLMs in text expansion, we need further optimization. Second, the performance gap between supervised methods and our ICL-based approach appears to diminish with increasing training dataset size, suggesting the need for further analysis.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (grant no. 62176180). This work is supported by

Alibaba Group through Alibaba Research Intern Program, the National Natural Science Foundation of China (Grant Nos. 62176180) and the Shenzhen College Stability Support Plan (Grant Nos. GXWD20231130140414001).

References

Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *22nd International World Wide Web Conference, WWW '13*, pages 13–24. https://doi.org/10.1145/2488388.2488391

Rami Aly, Steffen Remus, and Chris Biemann. 2019. Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 – August 2, 2019, Volume 2: Student Research Workshop*, pages 323–330. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-2045

Rohan Bhambhoria, Lei Chen, and Xiaodan Zhu. 2023. A simple and effective framework for strict zero-shot hierarchical classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9–14, 2023*, pages 1782–1792. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-short.152

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. https://doi.org/10.18653/v1/2021.acl-long.337

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2022. Contrastnet: A contrastive learning framework for few-shot text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10492–10500. https://doi.org/10.1609/aaai.v36i10.21292

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.58

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese BERT. *arXiv preprint arXiv:1906.08101*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? Language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9–14, 2023*, pages 4005–4019. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.247

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. OpenPrompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-demo.10

Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2023. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023*. OpenReview.net.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335. https://doi.org/10.18653/v1/2022.acl-long.26

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9–14, 2023*, pages 14014–14031. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.783

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 3816–3830. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.295, PubMed: 34156278

Sanghun Im, Gibaeg Kim, Heung-Seon Oh, Seongung Jo, and Donghwan Kim. 2023. Hierarchical text classification as sub-hierarchy sequence generation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7–14, 2023*, pages 12933–12941. AAAI Press. https://doi.org/10.1609/aaai.v37i11.26520

Ke Ji, Yixin Lian, Jingsheng Gao, and Baoyuan Wang. 2023. Hierarchical verbalizer for few-shot hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2918–2933, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.164

Qiao Jin, Andrew Shin, and Zhiyong Lu. 2023. LADER: Log-augmented dense retrieval for biomedical literature search. In *Proceedings of SIGIR 2023*, pages 2092–2097. ACM. https://doi.org/10.1145/3539618.3592005

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*. arXiv:1412.6980v9.

Daphne Koller and Mehran Sahami. 1997. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8–12, 1997*, pages 170–178. Morgan Kaufmann.

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 *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18–21, 2017*, pages 364–371. https://doi.org/10.1109/ICMLA.2017.0-134

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.256

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen.

2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pages 100–114. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.deelio-1.10

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical text classification with reinforced label assignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 445–455. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1042

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7–11, 2022*, pages 11048–11064. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.759

Shengwen Peng, Ronghui You, Hongning Wang, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2016. DeepMeSH: Deep semantic representation for improving large-scale mesh indexing. *Bioinformatics*, 32(12):70–79. https://doi.org/10.1093/bioinformatics/btw294, PubMed: 27307646

Thomson Reuters. 2012. Web of science.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.191

Debaditya Shome and Kuldeep Yadav. 2023. Exnet: Efficient in-context learning for data-less text classification. *CoRR*, abs/2305.14622.

Koustuv Sinha, Yue Dong, Jackie Chi Kit Cheung, and Derek Ruths. 2018. A hierarchical neural attention-based text classifier. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 817–823, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1094

Junru Song, Feifei Wang, and Yang Yang. 2023. Peer-label assisted hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9–14, 2023*, pages 3747–3758. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.207

Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*, pages 819–862. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.60

Roger A. Stein, Patrícia Augustin Jaques, and João Francisco Valiati. 2019. An analysis of hierarchical text classification using word embeddings. *Information Sciences*, 471:216–232. https://doi.org/10.1016/j.ins.2018.09.001

Mengxuan Sun, Jinghao Niu, Xuebing Yang, Yifan Gu, and Wensheng Zhang. 2023. CEHMR: Curriculum learning enhanced hierarchical multi-label classification for medication

recommendation. *Artificial Intelligence in Medicine*, 143:102613. https://doi.org/10.1016/j.artmed.2023.102613, PubMed: 37673560

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566. The Association for Computer Linguistics. https://doi.org/10.3115/v1/P15-1150

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. GPT-NER: Named entity recognition via large language models. *CoRR*, abs/2304.10428.

Yue Wang, Dan Qiao, Juntao Li, Jinxiong Chang, Qishen Zhang, Zhongyi Liu, Guannan Zhang, and Min Zhang. 2023b. Towards better hierarchical text classification with data generation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9–14, 2023*, pages 7722–7739. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.489

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. https://doi.org/10.18653/v1/2022.acl-long.491

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. https://doi.org/10.18653/v1/2022.emnlp-main.246

Jiawei Wu, Wenhan Xiong, and William Yang Wang. 2019. Learning to learn and predict: A meta-learning approach for multi-label classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 4353–4363. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1444

Jie Xiong, Li Yu, Xi Niu, and Youfang Leng. 2023. XRR: Extreme multi-label text classification with candidate retrieving and deep ranking. *Information Sciences*, 622:115–132. https://doi.org/10.1016/j.ins.2022.11.158

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

Yue Yu, Yuchen Zhuang, Rongzhi Zhang, Yu Meng, Jiaming Shen, and Chao Zhang. 2023. Regen: Zero-shot text classification via training data generation with progressive dense retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9–14, 2023*, pages 11782–11805. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.748

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*. https://doi.org/10.48550/arXiv.2210.02414

Jiali Zeng, Fandong Meng, Yongjing Yin, and Jie Zhou. 2023. TIM: Teaching large language models to translate with comparison. *CoRR*, abs/2307.04408.

Haoxing Zhang, Xiaofeng Zhang, Haibo Huang, and Lei Yu. 2022a. Prompt-based meta-learning for few-shot text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1357, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.87

Ruohong Zhang, Yau-Shian Wang, Yiming Yang, Donghan Yu, Tom Vu, and Likun Lei. 2023. Long-tailed extreme multi-label text classification by the retrieval of generated pseudo label descriptions. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2–6, 2023*, pages 1062–1076. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-eacl.81

Xinyi Zhang, Jiahao Xu, Charlie Soh, and Lihui Chen. 2022b. LA-HCN: Label-based attention for hierarchical multi-label text classification neural network. *Expert Systems with Applications*, 187:115922. https://doi.org/10.1016/j.eswa.2021.115922

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022c. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.622

Fei Zhao, Zhen Wu, Liang He, and Xin-Yu Dai. 2023. Label-correction capsule network for hierarchical text classification. *IEEE ACM Transactions on Audio, Speech, and Language Processing*, 31:2158–2168. https://doi.org/10.1109/TASLP.2023.3282099

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18v24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. https://doi.org/10.48550/arXiv.2306.05685

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, ACL 2020, Online, July 5–10, 2020*, pages 1106–1117. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.104

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023*. OpenReview.net.