Meta-LMTC: Meta-Learning for Large-Scale Multi-Label Text Classification

Ran Wang, Xi'ao Su, Siyu Long, Xinyu Dai[*], Shujian Huang, Jiajun Chen
National Key Laboratory for Novel Software Technology, Nanjing University, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China
{wangr,nlp_suxa,longsy}@smail.nju.edu.cn
{daixinyu,huangsj,chenjj}@nju.edu.cn

Abstract

Large-scale multi-label text classification (LMTC) tasks often face long-tailed label distributions, where many labels have few or even no training instances. Although current methods can exploit prior knowledge to handle these few/zero-shot labels, they neglect the meta-knowledge contained in the dataset that can guide models to learn with few samples. In this paper, for the first time, this problem is addressed from a meta-learning perspective. However, the simple extension of meta-learning approaches to multi-label classification is sub-optimal for LMTC tasks due to long-tailed label distribution and coexisting of few- and zero-shot scenarios. We propose a meta-learning approach named META-LMTC. Specifically, it constructs more faithful and more diverse tasks according to well-designed sampling strategies and directly incorporates the objective of adapting to new low-resource tasks into the meta-learning phase. Extensive experiments show that META-LMTC achieves state-of-the-art performance against strong baselines and can still enhance powerful BERTlike models.

1 Introduction

Large-scale multi-label text classification (LMTC) is a fundamental and practical task in natural language processing (Tsoumakas et al., 2010). LMTC can be found in several domains, such as organizing documents in Wikipedia articles (Partalas et al., 2015), annotating medical records with diagnostic and procedure labels (Yan et al., 2010; Rios and Kavuluru, 2018), assigning legislation with relevant legal concepts (Chalkidis et al., 2019). Different from multi-class classification, the LMTC task aims to assign multiple labels from a large predefined set (typically thousands) to each instance.

Due to the large predefined label set and limited annotated resources, LMTC tasks usually face the challenges of long-tailed label distribution, i.e., many labels have few or even no annotated samples. For example, in EURLEX15K (Chalkidis et al., 2019), about 70% of seen labels have been assigned to less than 20 documents (i.e., few-shot labels); and more than 40% of the predefined labels are not associated with any document (i.e., zero-shot labels). To make matter worse, new labels continually emerge as the field evolves. Though few/zero-shot labels may not contribute heavily to the overall performance, correct prediction of such labels is crucial in some cases (Rios and Kavuluru, 2018). For instance, when assigning the diagnosis labels to electronic health records, incorrect predictions of these labels either bring unnecessary financial burdens or make patients ignore potential health risks. These factors require models to utilize few or no samples to accurately assign labels.

To cope with these few/zero-shot labels, current models typically match texts to feature vectors for each label obtained by *exploiting prior label information*. Specifically, Rios and Kavuluru (2018) utilizes label textual descriptors to generate a feature vector for each label. Also, it employs a 2-layer graph convolutional neural network (Kipf and Welling, 2017) to take advantage of the structured knowledge of label spaces to enhance label representations. Apart from that, Lu et al. (2020) finds that label similarity graphs based on pre-trained word embeddings and co-occurrence frequency are also beneficial.

Nonetheless, these approaches neglect the potential meta-knowledge contained in the dataset that can guide the models to learn with only a small amount of samples. Meta-learning has been suggested as an efficient strategy to acquire this knowledge. To acquire meta-knowledge, meta-learning constructs the tasks of few-shot learning scenarios and aims to learn how to achieve maximum performance by utilizing a limited amount of samples (Vinyals et al., 2016; Snell et al., 2017). Following the idea of meta-learning (Qian and Yu,

[*] Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8633–8646
November 7–11, 2021. ©2021 Association for Computational Linguistics

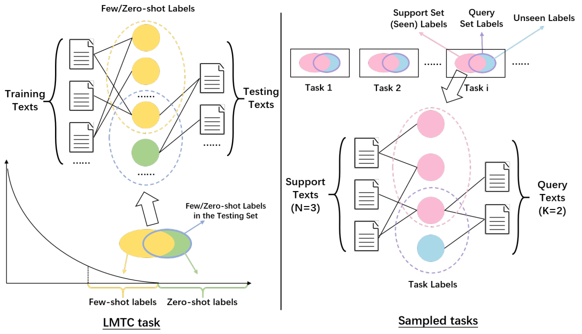

Figure 1: Illustration of the idea of META-LMTC. The left part shows models how to handle few/zero-shot labels in the LMTC task; the right part shows the simulated low-resource multi-label classification tasks, where the numbers of instances in the support set and in the query set are $N = 3$ and $K = 2$, respectively.

2019), for the first time, we propose to investigate the problem of few/zero-shot labels in LMTC tasks from a meta-learning perspective. Illustrated in Fig. 1, we simulate some few/zero-shot scenarios, which are faithful to the LMTC task and thereby provide chances for models to learn how to adapt fast and efficiently with a limited amount of data.

However, most meta-learning algorithms are designed for *multi-class* classification under the few-shot setting (Vinyals et al., 2016), and it is critical for meta-learned models' generalization to construct faithful and diverse tasks (Snell et al., 2017; Bansal et al., 2020). We argue that the simple extension of these approaches to *multi-label* classification is sub-optimal for the LMTC tasks in that (1) LMTC tasks need to cope with few- and zero-shot scenarios, while existing methods only consider few-shot ones, which is not faithful to the LMTC tasks. (2) LMTC tasks often face the challenge of long-tailed data distribution. However, these algorithms are not designed for specific data distribution and thereby makes the rare labels in the training set less involved in the meta-learning process, which reduces the diversity of the tasks.

To address the above two issues, we propose an optimization-based meta-learning algorithm, namely META-LMTC, which contains the meta-learning phase and fine-tuning phase. We design a task sampling strategy when considering the characteristics of LMTC tasks (i.e., the coexistence of few- and zero-shot scenarios, long-tailed data distribution). During the meta-learning phase, this strategy not only constructs more faithful meta-learning tasks (i.e., the zero- and few-shot scenarios coexist) but also provides more diverse labels and more various instances. Then the model acquires meta-knowledge on these tasks through the alternating meta-training process and meta-evaluation process. During the fine-tuning phase, the meta-learned model is fine-tuned on the original LMTC dataset to further improve performance. In summary, our contributions are as following:

- We propose a meta-learning algorithm META-LMTC for LMTC tasks. To our best knowledge, we are the first study to address these challenges in LMTC tasks from the meta-learning point of view.
- Our method outperforms the current state-of-the-art models on two LMTC benchmarks. Further analysis reveals that our method can still enhance powerful BERTlike models.

2 Related Work

Our work is a synthesis of two research directions: large-scale multi-label text classification and meta-learning. We review them in this section.

2.1 Large-Scale Multi-Label Text Classification

The skewed label frequency distribution of LMTC datasets poses few/zero-shot challenges for current models. Leveraging prior knowledge about labels has become a promising approach of tackling these problems. Rios and Kavuluru (2018) utilizes label descriptors and hierarchy to generate a representation for each label, with promising results. To further enhance these rare label representations, Lu et al. (2020) fuses pre-defined word embeddings and label co-occurrence graphs. Additionally, some studies find that a more powerful text encoder can improve the performance of frequent labels (Chalkidis et al., 2019, 2020; Li and Yu, 2020). Different from these existing solutions, we directly tackle the few/zero-shot label learning challenges from a meta-learning perspective.

2.2 Meta-Learning

Meta-learning (a.k.a. learning-to-learn) aims to learn a general model that can quickly adapt to a new task given a limited amount of annotated instances without suffering from overfitting (Geng et al., 2019). Most recent approaches to meta-learning focus on few-shot learning, which can be broadly categorized into (i) metric- (Vinyals et al., 2016; Snell et al., 2017) (ii) model- (Santoro et al., 2016; Ravi and Larochelle, 2017) and (iii) optimization-based techniques (Finn et al., 2017; Yoon et al., 2018). Meta-learning has been applied in various circumstances, such as image classification (Finn et al., 2019; Rajeswaran et al., 2019), machine translation (Gu et al., 2018), dialogue systems (Mi et al., 2019; Qian and Yu, 2019), etc.

Different from the above studies on multi-class classification under the few-shot setting, our work focus on LMTC tasks, where one document may be assigned multiple labels from a large predefined label set. In this work, we propose META-LMTC that is more suitable for encouraging general and robust representation in LMTC tasks. Unlike the few-shot learning that only focuses on the performance of novel classes, the LMTC tasks are concerned with the performances of all labels (including few/zero ones). To the best of our knowledge, we are the first to frame LMTC as a meta-learning problem.

3 Perliminaries

3.1 Large-Scale Multi-Label Text Classification

As mention before, LMTC tasks face a serious long-tailed problem, often involve few/zero-shot labels. Formally, we have two disjoint sets of seen labels $\mathcal{C}_S$ and unseen (i.e., zero-shot) labels $\mathcal{C}_U$. According to the label frequency, $\mathcal{C}_S$ can be further divided into frequent labels $\mathcal{C}_S^R$ and few-shot labels $\mathcal{C}_S^F$ such that $\mathcal{C}_S^R \cup \mathcal{C}_S^F = \mathcal{C}_S$ and $\mathcal{C}_S^R \cap \mathcal{C}_S^F = \emptyset$. Given a training set $D^{tr} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|D^{tr}|}$, where $\mathbf{x}_i$ indicates the i-th document and $\mathbf{y}_i \subset \mathcal{C}_S$ is the corresponding labels of $\mathbf{x}_i$, our goal is to predict correct labels $\hat{\mathbf{y}} \subset \mathcal{C}_S \cup \mathcal{C}_U$ for each testing document. Apart from training and testing set, some prior knowledge of labels, such as label descriptions, predefined label hierarchy is also available.

3.2 Model-Agnostic Meta-Learning

Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) is an optimization-based meta-learning framework. Its core idea is to leverage a set of auxiliary tasks to search for a good parameter initialization from which learning a target task would require only a handful of training samples. Formally, MAML first *meta-learns* the initialization of model parameters θ_0 with auxiliary tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_i\}$ and continue to *learn* the optimized parameters θ^* for target task $\mathcal{T}_t$ (Gu et al., 2018):

$$\theta^* = \text{Learn}(\mathcal{T}_t; \text{MetaLearn}(\mathcal{T}_1, \cdots, \mathcal{T}_i; \theta_0))$$

Notably, the original MAML is designed for few-shot multi-class classification problems and does not consider specific data distributions. However, due to the coexistence of few- and zero-shot scenarios and long-tailed data distribution, a simple extension of MAML for multi-label classification problems can be sub-optimal to LMTC tasks.

4 Meta-Learning for LMTC

In this section, we first define LMTC tasks from the meta-learning perspective. Then we present a detailed description of the proposed META-LMTC.

4.1 Problem Statement

Previous studies formulate the LMTC tasks as a traditional supervised learning process $\text{Learn}(\mathcal{T}_{\text{LMTC}}; \theta_0)$, where initial parameters θ_0 are obtained either randomly or pre-trained. Instead, from meta-learning perspective, we aim to find a better initialization θ_0^* with auxiliary low-resource multi-label text classification tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_i\}$, i.e., $\theta_0^* = \text{MetaLearn}(\mathcal{T}_1, \cdots, \mathcal{T}_i; \theta_0)$. In each $\mathcal{T}_i$, a support set $D_{\mathcal{T}_i}^{tr} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ and a query set $D_{\mathcal{T}_i}^{val} = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^K$ are sampled from the LMTC training data D^{tr} with a specific strategy τ, where $D_{\mathcal{T}_i}^{tr} \cap D_{\mathcal{T}_i}^{val} = \emptyset$ and N, K are the number of instances in the support set and query set. In addition, we let $\mathcal{C}_{\mathcal{T}_i}^{tr} = \bigcup_{n=1}^N \mathbf{y}_n$ and $\mathcal{C}_{\mathcal{T}_i}^{val} = \bigcup_{k=1}^K \mathbf{y}_k$ be the corresponding labels of support set and query set in $\mathcal{T}_i$. As far as we know, it is the first attempt to cope with few/zero-shot labels in LMTC from the perspective of meta-learning.

4.2 Overview of META-LMTC

Algo. 1 shows an overall procedure of META-LMTC, which consist of a meta-learning phase and a fine-tuning phase. We describe the meta-learning stage in detail here. Suppose we are given a model f_θ with parameters θ and a task sampling strategy τ which generates tasks $\mathcal{T}_i$. For each task $\mathcal{T}_i = (D_{\mathcal{T}_i}^{tr}, D_{\mathcal{T}_i}^{val})$, we first update the model parameters using one-step gradient descent as

$$\theta_i' = \theta - \alpha \nabla_\theta L_\theta(D_{\mathcal{T}_i}^{tr}) \qquad (1)$$

where α is the local learning rate and L is the loss function. After that, the loss of local parameters on the corresponding query set is computed, i.e., $L_{\theta_i'}(D_{\mathcal{T}_i}^{val})$. Finally, the global parameters are obtained using the loss across multiple tasks, i.e.,

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{D_{\mathcal{T}_i}^{val}} L_{\theta_i'}(D_{\mathcal{T}_i}^{val}) \qquad (2)$$

Algorithm 1 META-LMTC

```
Input: Dataset $D$, learning rates $\alpha, \beta$ and task sampling strategy $\tau$
Output: Model $\theta^*$
Initialize parameters $\theta = \theta_0$
// Meta-Learning Phase
while not done do
    Simulate a batch of low-resource multi-label text classification tasks $\mathcal{T}_i$ using strategy $\tau$
    for all $\mathcal{T}_i = (D^{tr}_{\mathcal{T}_i}, D^{val}_{\mathcal{T}_i})$ do
        Compute local parameters $\theta'_i$ with Eq. 1
    end for
    Update global model parameters $\theta$ with Eq. 2
end while
// Fine-tuning Phase
Fine-tune the model initialized with meta-learned parameter $\theta^*_0$ on the dataset $D$
Return the final model $\theta^*$
```

where β is the global learning rate. In short, META-LMTC explicitly simulates the low-resource LMTC tasks, and directly incorporates the objective of adapting to these tasks into the meta-learning optimization phases. This encourages models to learn meta knowledge, i.e., how to obtain maximal performance on these rare/unseen labels with little training data.

However, how to construct tasks is one of the main challenges for meta-learning (Vinyals et al., 2016). Snell et al. (2017) pointed out that a more faithful training problem to the test environment can lead to better performance and Bansal et al. (2020) claimed that the diversity in tasks for meta-learning is beneficial for models' generalization ability. Thus, the design of task sampling strategy τ to make tasks faithful and diverse is a critical problem to be solved.

4.3 LMTC Task Sampling Strategy

As mentioned before, a simple extension of meta-learning algorithms for the multi-label classification problem is sub-optimal for LMTC tasks in two following issues: (1) LMTC tasks need to cope with few- and zero-shot scenarios, while existing methods are only considering few-shot scenarios and thereby provide less faithful training condition; (2) LMTC datasets often exhibit long-tailed distribution. But meta-learning algorithms do not take into account the specific data distribution. The constructed tasks by their naive task sampling strategy just consider a limited amount of frequent labels more and reduce the diversity of the tasks.

To address the issue (1), we design a simple yet effective task sampling strategy, namely *instance-based* one: a handful of samples are uniformly sampled from the original LMTC dataset D and partitioned into two disjoint set, i.e., the support set $D^{tr}_{\mathcal{T}_i} = \{(\mathbf{x}_n, \mathbf{y}_n) | \mathbf{y}_n \subset \mathcal{C}^{tr}_{\mathcal{T}_i}\}^N_{n=1}$ and the query set $D^{val}_{\mathcal{T}_i} = \{(\mathbf{x}_k, \mathbf{y}_k) | \mathbf{y}_k \subset \mathcal{C}^{val}_{\mathcal{T}_i}\}^K_{k=1}$. We have found empirically that this strategy can construct more faithful tasks in which few- and zero-shot scenarios coexist, i.e., $\mathcal{C}^{val}_{\mathcal{T}_i} \cap \mathcal{C}^{tr}_{\mathcal{T}_i} \neq \emptyset$ and $\mathcal{C}^{val}_{\mathcal{T}_i} - \mathcal{C}^{tr}_{\mathcal{T}_i} \neq \emptyset$, with a high probability.[1] However, this strategy is still affected by the long-tailed label distribution of LMTC, as shown in the upper part of Fig. 2: the few-shot labels in the training set have fewer chance to appear in the tasks and models are more susceptible to meta-overfitting (Bansal et al., 2020) to a handful of frequent labels.

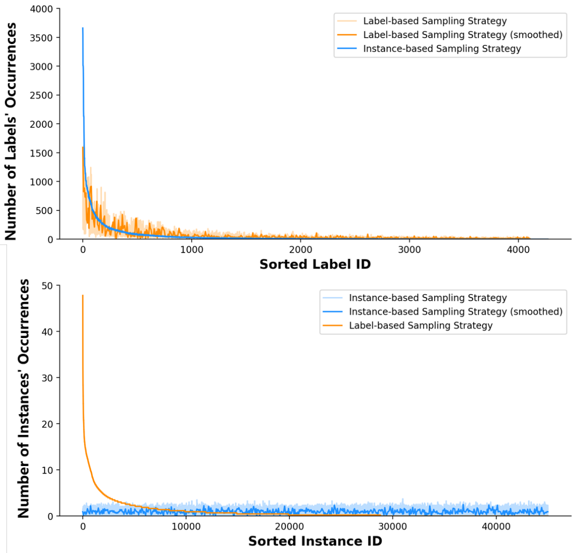

Figure 2: The number of times each label and instance occurring in the tasks sampled from EURLEX57K according to the instance- or label-based sampling strategy. Labels are sorted by the number of occurrences in the training set in descending order. Instances are sorted in descending order by the number of being sampled in the label-based sampling algorithm. Some curves are smoothed for clarity.

To alleviate this issue (issue (2)), we provide another strategy, namely *label-based* one: a label is first sampled from the label space $\mathcal{C}_S$, and then an instance annotated with this label is selected. We repeat this process $N + K$ times to construct $\mathcal{T}_i = (D^{tr}_{\mathcal{T}_i}, D^{val}_{\mathcal{T}_i})$. The upper part of Fig. 2 shows that the label-based one is fairer than the instance-based one from the label dimension. On the other hand,

[1] We sampled 10,000 tasks on the MIMIC-III and EURLEX57 datasets respectively using this strategy (where $N = 128$ and $K = 32$). *All* of these sampled tasks provided both few- and zero-shot scenarios. Statistically, on the MIMIC-III (or EURLEX57K) dataset, about 38.49% (or 46.50%) of the labels in the query set were unseen in the support set.

the lower part of Fig. 2 reveals that the instance-based one shows no biases to instances, while the label-based one pays too much attention to those instances mostly annotated with few-shot labels.

Though both the instance- and the label-based strategies provide more faithful tasks, they reduce diversity in the tasks from either the label dimension or the instance dimension. To increase diversity in tasks for meta-learning, we use a sampling ratio $p \in [0, 1]$ and each task $\mathcal{T}_i$ is constructed by the instance-based strategy with probability p or by the label-based one with probability $1 - p$. By appropriately setting the value of p, META-LMTC can provide more faithful and more diverse tasks and thereby boost models' performance.

5 Experiments

In this section, we conduct several experiments to evaluate the efficacy of our method in LMTC tasks. The experimental result shows that our method can bring performance improvements to all of the few/zero-shot LMTC base models.

5.1 Datasets

To evaluate our method, we use two benchmarks, a medical dataset MIMIC-III [2] (Johnson et al., 2016) and a EU legislation dataset, EURLEX57K (Chalkidis et al., 2019). [3] **MIMIC-III** contains approximately 58k English discharge summaries from US hospitals. Each summary is annotated with codes (labels) from 6,966 leaves of the ICD-9 diagnosis hierarchy, with an average of 11 labels. Another benchmark **EURLEX57K** are the LMTC dataset in the legal domain, which contains 57k English legislative documents. Each document is annotated with an average of five concepts (labels) from the 4,271 concepts of EUROVOC[4].

Following Rios and Kavuluru (2018); Lu et al. (2020), the labels are divided into *frequent*, *few-shot* and *zero-shot* labels. Specifically, few-shot labels are defined as those whose frequencies in the training set are less than or equal to 5 for MIMIC-III and 50 for EURLEX57K[5]. In addition, MIMIC-III does not contain a standardized training/test split. We create our split that ensures the same patient does not appear in both the training and test datasets. The ICD-9 diagnosis codes in MIMIC-III are the labels to be assigned. Detailed statistics of these datasets are shown in Table 1.

	Dataset	MIMIC-III	EURLEX57K
Doc	# Train	46,562	45,000
	# Dev	5,829	6,000
	# Test	5,970	6,000
	Avg # labels	11	5
Label	# Frequent	3,282	746
	# Few	3,344	3,362
	# Zero	340	163

Table 1: Dataset statistics

5.2 Evaluation Metrics

Because there are thousands of labels in LMTC datasets, annotators or users would see a label unless it appears at the top of the ranking. Thus, ranking metrics are usually adopted to measure the usefulness of various systems (Rios and Kavuluru, 2018; Lu et al., 2020; Chalkidis et al., 2020). Following them, we report both recall at k (R@K) and normalized discounted cumulative gain at k (nDCG@K), where K is set to 10 for MIMIC-III and 5 for EURLEX57K. Because our aim is high performance on both frequent and few/zero-shot labels, similar to the setup in Xian et al. (2019) and Rios and Kavuluru (2018), we also report the harmonic average across all R@K and all nDCG@K scores for methods that can predict zero-shot labels.

5.3 Baselines

Following Lu et al. (2020), we compare the following baselines.

CNN (Kim, 2014) uses convolutional neural networks with max-pooling to extract text features, which are then used to make the predictions for the labels.

RCNN (Lai et al., 2015) uses recurrent neural networks with a convolution layer to consider both long-distance and local dependencies. It achieves best the performances across competitive text encoders in Liu et al. (2019).

CAML (Mullenbach et al., 2018) is a model designed for clinical notes and text documents. It

[2] https://physionet.org/content/mimiciii/1.4/

[3] According to the data maintainer (https://github.com/MIT-LCP/mimic-code/issues/898), MIMIC-II is no longer available. We cannot include it in experiments.

[4] http://eurovoc.europa.eu

[5] 50 seems too high for few-shot labels. But we follow the setting of existing works to make the performance of our implementations comparable to reported performances. Furthermore, Fig. 3 shows that our method can bring even more improvement with a smaller threshold. Thus, we believe that a smaller threshold would not affect the conclusions of our experiments.

MIMIC-III	Frequent		Few-shot		Zero-shot		Harmonic Average	
	R@10	nDCG@10	R@10	nDCG@10	R@10	nDCG@10	R@10	nDCG@10
CNN	34.6	44.2	5.5	2.9	-	-	-	-
RCNN	43.9	56.0	14.2	9.8	-	-	-	-
CAML	41.2	53.3	5.9	3.9	-	-	-	-
ZAGRU	49.0	61.3	26.9	17.7	34.7	22.2	34.7	25.5
+ SIMPLE-EXT	49.2	61.8	27.2	17.8	35.4	23.5	35.2	26.1
+ META-LMTC	49.7*	62.6*	29.1*	20.2*	38.8*	24.1	37.4*	28.0*
ZAGGRU	49.1	61.8	26.2	17.4	33.9	24.1	34.1	26.1
+ SIMPLE-EXT	49.4	62.0	27.8	18.7	36.0	24.4	35.7	27.1
+ META-LMTC	49.6*	62.3*	28.3*	19.7*	40.9*	25.2	37.5*	28.2*
AGRU-KAMG	49.5	62.4	28.9	19.1	34.3	23.0	35.7	26.8
+ SIMPLE-EXT	50.0	63.2	29.2	19.8	38.3	25.4	37.3	28.4
+ META-LMTC	**50.2***	**63.2***	**32.7***	**22.3***	**43.3***	**28.4***	**40.8***	**31.3***

EURLEX57K	Frequent		Few-shot		Zero-shot		Harmonic Average	
	R@5	nDCG@5	R@5	nDCG@5	R@5	nDCG@5	R@5	nDCG@5
CNN	71.8	78.2	56.5	50.8	-	-	-	-
RCNN	68.8	75.3	53.3	47.6	-	-	-	-
CAML	66.2	72.7	43.5	39.2	-	-	-	-
ZAGRU	70.9	77.1	56.2	51.9	51.1	40.9	58.3	52.9
+ SIMPLE-EXT	73.2	79.6	60.3	55.8	53.9	41.2	61.5	54.8
+ META-LMTC	74.2*	80.6*	65.3*	60.0*	57.9*	45.3*	65.1*	58.7*
ZAGGRU	71.9	78.2	56.9	51.8	50.6	41.6	58.5	53.4
+ SIMPLE-EXT	72.1	78.6	57.7	53.2	52.2	40.2	59.6	53.2
+ META-LMTC	**75.2***	**81.6***	**65.5***	**60.6***	56.7*	45.7*	64.9*	**59.2***
AGRU-KAMG	72.4	78.9	59.1	54.2	54.5	43.7	61.1	55.5
+ SIMPLE-EXT	72.8	79.2	60.7	55.4	53.4	42.7	61.3	55.5
+ META-LMTC	74.2*	80.6*	64.3*	59.4*	**59.0***	**46.3***	**65.2***	59.0*

Table 2: Results (%) of experiments across all the methods for frequent, few-shot, and zero-shot label groups. The first three methods are incapable of zero-shot learning. Bold figures are the best results for each metric among all the methods considering the zero-shot problems. SIMPLE-EXT denotes the simple extension of MAML for the multi-label classification problem. The corresponding p values for the ZAGRU, ZAGGRU, AGRU-KAMG equipped with META-LMTC are 0.5, 0.67, 0.5 on the MIMIC-III and 0.67, 0.67, 0.33 on the EURLEX57K respectively. Additionally, * indicates META-LMTC achieves significantly improvement on the base model (pairwise t-test at 0.05 significance level).

uses the label-wise attention mechanism, allowing each label to focus on different parts of the text.[6]

ZAGGRU (Chalkidis et al., 2019) originally proposed by Rios and Kavuluru (2018), applies graph convolutions (GCNs) to the label hierarchy.[7] Its GCNs can obtain better representations for few/zero-shot labels benefit from the (better) representations of frequent labels that are nearby in the label hierarchy.

ZAGRU is an ablation method of ZAGGRU proposed in Chalkidis et al. (2020). ZAGRU replaces the stack of GCN layers in ZAGGRU into a plain two-layer Multi-Layer Perceptron (MLP). The model though is unaware of the label hierarchies yet produces a surprisingly competitive performance of rare labels.

AGRU-KAMG (Lu et al., 2020) is the state-of-the-art model of LMTC task, which can handle few- and zero-shot labels. It utilizes the label graphs based on the similarity among labels' embeddings and the label co-occurrence graphs besides the predefined label hierarchy, which captures label relations from different views and thereby enhances the quality of labels' representations.

Among the above models, the first three use randomly initialized label embedding for each label, which results in their incapability of coping with unseen labels and poor generalization over rare labels. Instead, the last three models use a shared label encoder to obtain label representations, which empowers them to handle few/zero-shot labels. Because we focus on the models' generaliza-

[6] The original model uses a CNN text encoder whereas we use a Bi-GRU for better performance and fairness of comparison.

[7] According to Chalkidis et al. (2020), a Bi-GRU encoder can obtain better performance than the CNN token encoder of the original model. Thus, we use Bi-GRUs rather than CNNs as the token encoder.

tion over both frequent labels and few/zero-shot labels, META-LMTC is only applied to the last three models to verify its effectiveness and versatility. To explore the necessity of a balanced task sampling strategy, we also apply the simple extension of MAML for the multi-label classification problem (called SIMPLE-EXT[8]) to the same base models.

5.4 Implementation Details

We implement all the methods relying on the PyTorch library. We also use Higher (Grefenstette et al., 2019) for our meta-learners. Additionally, the binary cross-entropy loss is used as the loss function during the meta-training and fine-tuning phases. More details can be found in Appendix A.

5.5 Results

The experimental results of our methods and the baselines on the MIMIC-III and EURLEX57K datasets are shown in Table 2. We apply our framework to the ZAGRU, ZAGGRU, and AGRU-KAMG models. The performance of the models meta-trained by pure instance- and label-based strategies is not reported here due to space limitations but can be found in Appendix C.

As shown in the upper part of Table 2, the AGRU-KAMG model meta-trained with our method performs the best in every single evaluation metric among all of the models on the MIMIC-III dataset. Equipped with our method META-LMTC, the state-of-the-art model, ARGU-KAMG, has achieved relative improvements of 13.1% R@10 and 16.8% nDCG@10 on few-shot labels along with 26.2% R@10 and 23.5% nDCG@10 on zero-shot labels. In addition to the few/zero-shot labels, performance on frequent ones can also benefit from our method. We argue this is because our method can obtain better initialization for these frequent labels. It is worth noting that the performance of all of the three base models is significantly improved when equipped with META-LMTC, which verifies its versatility.

The lower part of Table 2 presents the results of our proposed methods and the baselines on the EURLEX57K testing set. Similar to the experimental results on the MIMIC-III dataset, our proposed methods still bring great improvement to all of the base models in each label group and outperform the baselines by a large margin. Specifically, by employing our method, the harmonic average nDCG@5 of ZAGRU, ZAGGRU, and AGRU-KAMG is absolutely improved by 5.8% and 5.8% and 3.5% respectively. This further confirms that our method is capable of helping each model to predict the labels more accurately.

Table 2 also shows the performance of the three base models equipped with SIMPLE-EXT on the dataset MIMIC-III and EURLEX57K. Although this method can boost models' performance, it is not as effective as our method in that SIMPLE-EXT neglects the zero-shot scenarios and long-tailed label distributions of LMTC datasets.

5.6 Analysis

We explore the following questions further in the section: Is META-LMTC also effective to the powerful BERTlike models? How does hyperparameters choice affect this method? Which labels benefit more from our method?

5.6.1 Apply to BERTlike Model

Most recently, Chalkidis et al. (2020) shows BERTlike models (Devlin et al., 2019) equipped with label-wise attention networks (BERT-LWAN) has the best results among all methods on EURLEX57K. But the BERT-LWAN relies solely on trainable vectors to represent labels and thereby cannot handle unseen labels. However, it is not trivial to extend this model to zero-shot scenarios. To cope with unseen labels, BERT-LWAN needs to employ a shared label encoder to encode each label's text description as its representation. Due to a large number of predefined label sets in LMTC datasets, it is impractical to use BERT as the shared label encoder.[9]

Fortunately, Sanh et al. (2019) presents a smaller, faster and lighter model called DistilBERT while retaining 97% of BERT's language understanding capabilities. Thus, we employ this

[8] Specifically, the support sets and the query sets are constructed like that of the instance-based task sampling strategy. But when computing the loss on the query set, only seen labels on the support sets are considered. In another word, SIMPLE-EXT only constructs few-shot scenarios during the meta-learning phases.

[9] We also tried to simply average the word embeddings of the label description, use a multi-layer perceptron or employ a graph neural network as the shared label-encoder like ZAGRU, ZAGGRU, and KAMG. But in this case, the model can not converge, which may be caused by the huge gap of expression ability between the text encoder and label encoder. We further used the BERT as the shared label encoder, but the out-of-memory issue was raised even using four 32G V100s. To solve the issue, the gradient accumulation trick was applied, but it needed more than ten days to converge; negative sampling in the label space can accelerate the training process, but the performance on frequent labels had a significant degradation.

	Harmonic Average	
	R@5	nDCG@5
Z-DistilBERT	70.14	66.15
+ META-LMTC	71.64	67.08

Table 3: Harmonic average metrics on EURLEX57K of Z-DistilBERT with or without META-LMTC.

model as the shared label encoder and equip it with LWAN and the gradient accumulation trick, which is dubbed as Z-DistilBERT capable of zero-shot learning. [10]

Table 3 shows the difference in various metrics on EURLEX57K using Z-DistilBERT with or without META-LMTC. It clearly demonstrates the META-LMTC can still bring significant improvement even to the powerful BERTlike model, Z-DistilBERT.

5.6.2 Hyperparameter Studies

The META-LMTC improves the generalization ability of models by increasing the diversity of meta-learning tasks and the task distribution depends on the hyperparameter p. In this subsection, we investigate the influence of this hyperparameter on the models' performances.

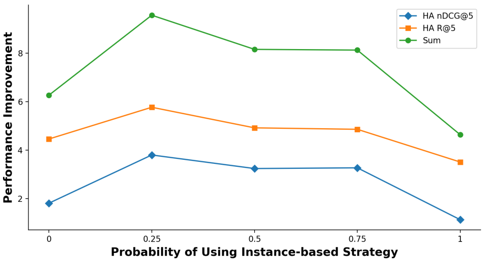

Figure 3: The performance improvements of ZAGGRU+META-LMTC with different hyperparameter p compared to the base model on the EURLEX57K.

Fig. 3 presents the difference between the performance of ZAGGRU equipped with META-LMTC and that of the base model. The hyperparameter p is chosen from $\{0.00, 0.25, 0.50, 0.75, 1.00\}$. Note that $p = 0.00$ is the pure label-based task sampling strategy while $p = 1.00$ is the pure instance-based one. It demonstrates that META-LMTC can consistently boost the performance of the base model with all different values of p. But the value of p can significantly affect the performance of META-LMTC. In general, the performance improves at first and then decreases as the value of p increases. As discussed before, pure task sampling strategies are inferior because they ignore the long-tailed distribution of label frequency in LMTC datasets and reduce the diversity of sampled tasks. Using other base models, the experimental results also show a similar trend, which can be found in Appendix D.

5.6.3 Performance Improvement Breakdown

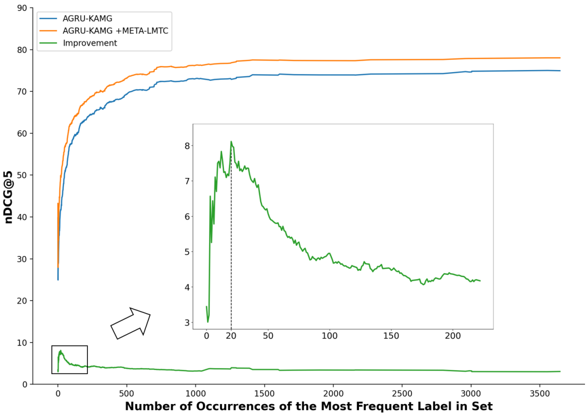

Figure 4: Performance for AGRU-KAMG with and without META-LMTC on the EURLEX57K dataset. The green line denotes the improvement of AGRU-KAMG with the addition of META-LMTC. Both models are evaluated with nDCG@5 for label sets with different maximum label frequencies (values on the x-axis).

To completely understand the source of the performance boost, we resort to a detailed performance improvement breakdown, presented in Fig. 4. The green line in this figure indicates the performance difference between applying or not applying META-LMTC to the base model when considering labels with a frequency less than or equal to a certain value. As can be seen, our method has the greatest benefit for zero-shot labels and few-shot labels whose frequencies are between 1 and 20. This reveals that META-LMTC does improve the models' ability to handle few/zero-shot labels.

6 Discussion

In this section, we report some experimental results of evaluating few- and/or zero-shot labels in the LMTC tasks in stricter settings.

6.1 Construction of the Zero-shot Label Candidates

When evaluating models' performance over unseen labels, existing works just consider only the labels appearing in the datasets (i.e., the validation or testing test) but not all available labels. However, in the realistic setting, we only know that the unseen label appears in the predefined label set. For that,

[10] This model takes about 2.5 days to converge on EURLEX57K. More details can be found in Appendix B.

we consider all the available labels but not appearing in the training set as zero-shot label candidates. Because the number of zero-shot labels are dramatically increases, the performances of all models drop dramatically. For example, the R@5 and nDCG@5 of the ZAGRU model drops from 54.5% and 43.7% to 20.7% and 14.6% respectively in the EURLEX57K dataset. But our method can still bring performance enhancement to these base models. When equipped with META-LMTC, they rise to 23.8% (+3.1%) and 16.4% (+1.8%).

6.2 Evaluation Metrics of the Few/Zero-shot Labels

In LMTC tasks, ranking-based metrics are often adopted to evaluate the top K labels with the highest scores predicted by the model, e.g. R@K and nDCG@K. In previous works, the value of K is selected based on the average number of labels per document. However, the average numbers of few- and/or zero-shot labels in each dataset are much lower than the selected K, which may lead to inappropriate evaluation on these labels. For example, the average numbers of few- and zero-shot labels in the EURLEX57K dataset are about 1.7 and 1.1 respectively (instead of 5), so we set K=2 and K=1 for few- and zero-shot evaluation. Under these settings, the performance of the AGRU-KAMG model on the few-shot labels becomes 52.5% R@2 and 57.3% nDCG@2. As for the zero-shot labels, AGRU-KAMG gets 24.1% R@1 and 25.8% nDCG@1. Even though the model's performance shows an obvious difference, our method can still bring steady improvement on both few- and zero-shot labels, specifically 55.2% R@2 (+2.7%) and 60.7% nDCG@2 (+3.4%) for few-shot labels along with 26.9% R@1 (+2.8%) and 28.3% nDCG@1 (+2.5%) for zero-shot ones.

7 Conclusion and Future Work

In this paper, we proposed an optimization-based meta-learning framework, namely META-LMTC, along with several task sampling strategies. We are the first study to address the LMTC tasks from a meta-learning perspective. Extensive experimental results showed that our method is able to significantly improve the performance of all the base models. The further analysis presented that our method is also applicable to the strong BERTlike model, and revealed the source of the performance boost our method brings. As future work, we will further explore the meta-learning approaches to handle the generalized zero-shot learning problem (GZSL) in the LMTC tasks.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by NSFC Projects (Nos. 61936012 and 61976114), the National Key RD Program of China (No. 2018YFB1005102).

References

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 522–534.

Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. An empirical study on large-scale multi-label text classification including few and zero-shot labels. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7503–7515.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 6314–6322.

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*, pages 4171–4186.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135.

Chelsea Finn, Aravind Rajeswaran, Sham M. Kakade, and Sergey Levine. 2019. Online meta-learning. In *Proceedings of the 36th International Conference on Machine Learning,*, pages 1920–1930.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3902–3911.

Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. 2019. Generalized inner loop meta-learning. *Computing Research Repository*, arXiv:1910.01727.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*. OpenReview.net.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2267–2273.

Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8180–8187. AAAI Press.

Liqun Liu, Funan Mu, Pengyu Li, Xin Mu, Jing Tang, Xingsheng Ai, Ran Fu, Lifeng Wang, and Xing Zhou. 2019. Neuralclassifier: An open-source neural hierarchical multi-label text classification toolkit. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 87–92. Association for Computational Linguistics.

Jueqing Lu, Lan Du, Ming Liu, and Joanna Dipnall. 2020. Multi-label few/zero-shot learning with knowledge aggregated from multiple label graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2935–2943.

Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 3151–3157.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1101–1111.

Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artières, George Paliouras, Éric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Gallinari. 2015. LSHTC: A benchmark for large-scale text classification. *Computing Research Repository*, arXiv:1503.08581.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2639–2649. Association for Computational Linguistics.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 113–124.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations*. OpenReview.net.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *Computing Research Repository*, arXiv:1910.01108.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 1842–1850.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 4077–4087.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis P. Vlahavas. 2010. Mining multi-label data. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook, 2nd ed*, pages 667–685. Springer.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 3630–3638.

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2251–2265.

Yan Yan, Glenn Fung, Jennifer G. Dy, and Rómer Rosales. 2010. Medical coding classification by leveraging inter-code relationships. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 193–202.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*, pages 7343–7353.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1):1–9.

A Additional Implementation Details of Main Experiments

We extract the vocabularies from both the documents in the training texts and the label descriptors. Each document is truncated at the length of 512 at the training and inference stage.

Hyperparameters are selected with the best nDCG@K of the zero-shot labels on the validation set. The search space of each hyperparameter is shown in Table 4.

For all the models implemented in experiments of the two datasets, the one-layer Bi-GRU with hidden dimension 100 is used to set up the RNN encoders and 200 filters with kernel size 10 is for the CNN encoders. The size of the GCNs' hidden states is set to 200. Additionally, We used 200-dimensional word embeddings pretrained on PubMed (Zhang et al., 2019) and GloVe (Pennington et al., 2014) for the MIMIC-III and EURLEX57K respectively. The dropout rate is set to 0.1 for the embedding layer and 0.5 for the last hidden layer for all the implemented models.

In the meta-training phase, the SGD optimizer with learning rate $\alpha = 3 \times 10^{-3}$ is used for each task's local update in the MIMIC-III and $\alpha = 1 \times 10^{-3}$ in the EURLEX57K. The Adam optimizer with learning rate $\beta = 3 \times 10^{-4}$ is used to update the global parameters in the MIMIC-III and $\beta = 1 \times 10^{-4}$ in the EURLEX57K. The size of the support set and the query set is 128 and 32 respectively. Besides, the model's global parameters are updated once using the average loss of 4 sampled tasks. At last, the meta-model is saved for the fine-tuning phase after being updated by 300 iterations, i.e., learning from 1200 sampled tasks.

In the training phase, the batch size of 64 is used for both of the datasets. When training a model from scratch, the learning rate is set to 1×10^{-3} for MIMIC-III and 3×10^{-4} for EURLEX57K. If fine-tuning a model that has been meta-trained, the learning rate is 3×10^{-4} and 1×10^{-4} for MIMIC-III and EURLEX57K respectively.

All experiments are run with one NVIDIA GPU V100. In Table 5, we report the size of the models and the elapsed training time.

B Implementation Details of Z-DistilBERT

We implement the Z-DistilBERT model similar to ZAGRU but replace both text encoder and label encoder with DistilBERT. Due to the thousands of labels in LMTC tasks, the memory overhead will become unacceptable if all the labels are encoded at the same time. To overcome this issue, we divide the labels into many blocks with small sizes, e.g. 256 labels per block. For each block, the loss of its labels and the gradients of the model parameters are firstly computed. Then the gradient of each parameter are accumulated and the computation graph except for the text encoding part will be freed manually. When all the blocks are processed serially, model parameters are updated with the accumulated gradients.

C Full Experiments Results

The experimental results of our methods and the baselines on dataset MIMIC-III and EURLEX57K are shown in Table 6. We apply our algorithm to the ZAGRU, ZAGGRU, and AGRU-KAMG models based on the instance-based (META-LMTC-IS), label-based (META-LMTC-LS), and final sampling strategies (META-LMTC). The results show that all the existing models can obtain significant improvements in performance when being meta-trained with our method, which illustrates the effectiveness and versatility of our methods. Additionally, the final strategy outperforms the pure instance- or label-based ones in most of the metrics.

D Additional Hyperparameter Studies

Fig. 5 and Fig. 6 present the performance improvements brought by META-LMTC to the ZAGRU and KAMG models. The hyperparameter p is chosen from $\{0.00, 0.25, 0.50, 0.75, 1.00\}$. Note that $p = 0.00$ is the pure label-based one while $p = 1.00$ is the pure instance-based task sampling strategy. It demonstrates that with all different value of p, META-LMTC can consistently boost the performance of the base model. But the value of p can significantly affect the performance of META-LMTC. Generally speaking, pure task sampling strategies are sub-optimal because they ignore the long-tailed distribution of label frequency in LMTC datasets and reduce the diversity of sampled tasks, which does harm to the models' performance.

Models			Train	
hidden (filter) size	feature dropout	embedding dropout	batch size	learning rate
{100, 150, 200}	{0,0.5}	{0,0.1,0.2}	{64,128}	{1e-4,3e-4,1e-3}
Meta-train				
Iterations	local learning rate	global learning rate	support set size	query set size
{100,200,300,400,500}	{3e-4,1e-3,3e-3}	{1e-4,3e-4,1e-3}	{64,128}	{32,64}

Table 4: Hyperparameter search space of the models, training and meta-training stage.

Methods	Trainable Parameters	Training Time (MIMIC-III/EURLEX57K)
CNN	29	1.5h/1h
RCNN	29	2h/1.5h
CAML	30	3h/2h
ZAGRU	28	2.5h/2.5h
ZAGRU+META-LMTC	28	3h/3h
ZAGGRU	29	2.5h/2.5h
ZAGGRU+META-LMTC	29	3h/3h
AGRU-KAMG	31	3h/3h
AGRU-KAMG+META-LMTC	31	3.5h/3.5h
Z-DistilBERT	66	71h/59h
Z-DistilBERT+META-LMTC	66	82h/67h

Table 5: Number of parameters in millions and training time for a single run reported for all examined methods on the MIMIC-III and EURLEX57K datasets.

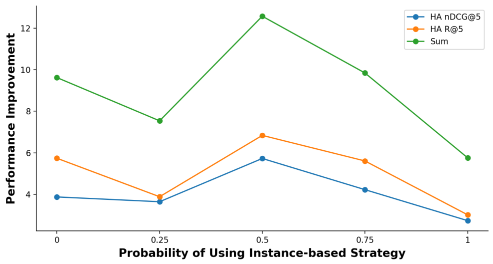

Figure 5: The performance improvements of ZAGRU+META-LMTC with different hyperparameter p compared to the base model on the EURLEX57K.

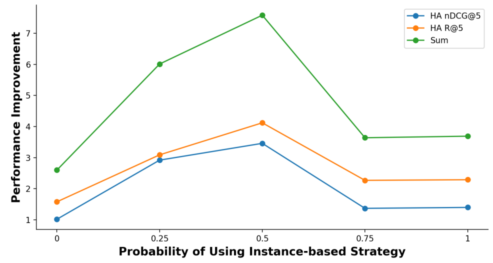

Figure 6: The performance improvements of AGRU-KAMG+META-LMTC with different hyperparameter p compared to the base model on the EURLEX57K.

MIMIC-III	Frequent		Few-shot		Zero-shot		Harmonic Average	
	R@10	nDCG@10	R@10	nDCG@10	R@10	nDCG@10	R@10	nDCG@10
CNN	34.6	44.2	5.5	2.9	-	-	-	-
RCNN	43.9	56.0	14.2	9.8	-	-	-	-
CAML	41.2	53.3	5.9	3.9	-	-	-	-
ZAGRU	49.0	61.3	26.9	17.7	34.7	22.2	34.7	25.5
+ META-LMTC-IS	49.4	62.2	26.9	18.6	36.6	24.3	35.4	27.0
+ META-LMTC-LS	49.2	61.8	27.0	18.8	35.9	23.6	35.2	26.9
+ META-LMTC	49.7	62.6	29.1	20.2	38.8	24.1	37.4	28.0
ZAGGRU	49.1	61.8	26.2	17.4	33.9	24.1	34.1	26.1
+ META-LMTC-IS	49.6	62.4	27.2	17.9	36.7	24.5	35.6	26.6
+ META-LMTC-LS	49.2	61.9	27.8	19.2	36.2	24.5	35.8	27.5
+ META-LMTC	49.6	62.3	28.3	19.7	40.9	25.2	37.5	28.2
AGRU-KAMG	49.5	62.4	28.9	19.1	34.3	23.0	35.7	26.8
+ META-LMTC-IS	50.0	63.0	29.2	19.9	40.1	27.0	37.9	29.1
+ META-LMTC-LS	**50.2**	63.1	32.6	21.6	39.8	26.2	39.6	29.9
+ META-LMTC	**50.2**	**63.2**	**32.7**	**22.3**	**43.3**	**28.4**	**40.8**	**31.3**

EURLEX57K	Frequent		Few-shot		Zero-shot		Harmonic Average	
	R@5	nDCG@5	R@5	nDCG@5	R@5	nDCG@5	R@5	nDCG@5
CNN	71.8	78.2	56.5	50.8	-	-	-	-
RCNN	68.8	75.3	53.3	47.6	-	-	-	-
CAML	66.2	72.7	43.5	39.2	-	-	-	-
ZAGRU	70.9	77.1	56.2	51.9	51.1	40.9	58.3	52.9
+ META-LMTC-IS	74.1	80.4	57.3	53.1	55.6	44.2	61.3	55.7
+ META-LMTC-LS	73.5	79.8	62.6	58.0	57.9	43.4	64.0	56.8
+ META-LMTC	74.2	80.6	65.3	60.0	57.9	45.3	65.1	58.7
ZAGGRU	71.9	78.2	56.9	51.8	50.6	41.6	58.5	53.4
+ META-LMTC-IS	**75.2**	**81.6**	58.6	53.7	55.6	41.5	62.1	54.6
+ META-LMTC-LS	73.0	79.5	62.0	56.2	56.2	41.8	63.0	55.3
+ META-LMTC	**75.2**	**81.6**	**65.5**	**60.6**	56.7	45.7	64.9	**59.2**
AGRU-KAMG	72.4	78.9	59.1	54.2	54.5	43.7	61.1	55.5
+ META-LMTC-IS	73.9	80.2	64.0	58.8	55.1	43.1	63.4	56.9
+ META-LMTC-LS	72.7	79.0	60.1	55.1	57.3	45.0	62.7	56.6
+ META-LMTC	74.2	80.6	64.3	59.4	**59.0**	**46.3**	**65.2**	59.0

Table 6: Results (%) of experiments across all the methods for frequent, few-shot, and zero-shot label groups. The first three methods are incapable of zero-shot learning. Bold figures are the best results for each metric among all the methods considering the zero-shot problems. The best results of each base model are shown underlined. The corresponding p values for the ZAGRU, ZAGGRU, AGRU-KAMG equipped with META-LMTC are 0.5, 0.67, 0.5 on the MIMIC-III and 0.67, 0.67, 0.33 on the EURLEX57K, respectively.