Disentangling Task Relations for Few-shot Text Classification via Self-Supervised Hierarchical Task Clustering

Juan Zha*[1]*, Zheng Li***[2], Ying Wei[3], Yu Zhang**†[4]
[1]University of Southern California, CA, USA
[2]Amazon.com Inc, CA, USA
[3]City University of Hong Kong, Hong Kong, China
[4]Southern University of Science and Technology, China
[1]juanzha@usc.com, [2]amzzhe@amazon.com
[3]yingwei@cityu.edu.hk, [4]yu.zhang.ust@gmail.com

Abstract

Few-Shot Text Classification (FSTC) imitates humans to learn a new text classifier efficiently with only few examples, by leveraging prior knowledge from historical tasks. However, most prior works assume that all the tasks are sampled from a single data source, which cannot adapt to real-world scenarios where tasks are heterogeneous and lie in different distributions. As such, existing methods may suffer from their globally knowledge-shared mechanisms to handle the task heterogeneity. On the other hand, inherent task relation are not explicitly captured, making task knowledge unorganized and hard to transfer to new tasks. Thus, we explore a new FSTC setting where tasks can come from a diverse range of data sources. To address the task heterogeneity, we propose a self-supervised hierarchical task clustering (SS-HTC) method. SS-HTC not only customizes cluster-specific knowledge by dynamically organizing heterogeneous tasks into different clusters in hierarchical levels but also disentangles underlying relations between tasks to improve the interpretability. Extensive experiments on five public FSTC benchmark datasets demonstrate the effectiveness of SS-HTC.

1 Introduction

Recent advances in deep learning highly rely on massive human annotations. This reliance increases the burden of data collection and meanwhile hinders its potentials to the low-data regime, where the labeled data is scarce and difficult to obtain. Inspired by human beings' capabilities that can quickly learn with a few examples, Few-Shot Learning (FSL) (Vinyals et al., 2016; Finn et al., 2017), which aims to learn a classifier that generalizes well even with a few training instances per class, has recently attracted much attention.

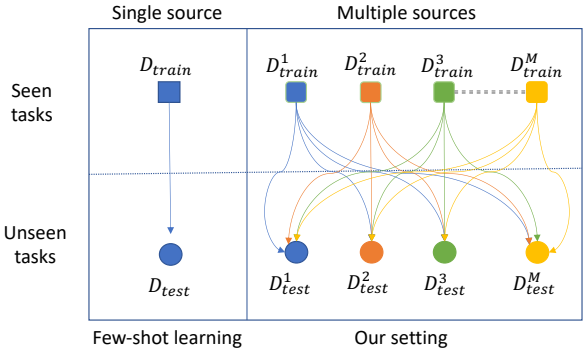


Figure 1: Comparison of existing FSTC formulation and our proposed practical problem setting.

In the NLP domain, Few-Shot Text Classification (FSTC) (Han et al., 2018) has been actively investigated in data-sparsity scenarios, e.g., relation classification (Han et al., 2018), event classification (Deng et al., 2020), and intent classification (Zhang et al., 2021), where new categories such as relations, events, or intent types tend to emerge frequently and lack sufficient annotations. Meta-learning (a.k.a. learning to learn) approaches (Finn et al., 2017), which transfer prior knowledge from previous tasks to improve the effectiveness in learning new tasks, have achieved superior performance for FSTC (Gao et al., 2019a; Sun et al., 2019; Bao et al., 2020). The prior knowledge can be instantiated as a transferable metric space for retrieving nearest prototypes in (Gao et al., 2019a; Sun et al., 2019), dynamic capsules in (Geng et al., 2019), and distributional signatures in (Bao et al., 2020), etc.

Despite their early success, such approaches have two main drawbacks: (1) they assume that all previous tasks are sampled from a single data source or domain, leading to tasks with low inter-task variance. As a consequence, these methods globally share the prior knowledge across all tasks but fail to handle real-world applications where the historical tasks that potentially contribute may come from diverse data sources in different distri-

*Most of the work was done when the first author was a research assistant at Southern University of Science and Technology; * Equal contribution; †Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5265 - 5276
December 7-11, 2022

butions (a.k.a. task heterogeneity (Vuorio et al., 2018)). For example, the knowledge learned from categorizing different types of products or services may be hardly transferred to classify public comments or opinions among different topics, or to determine users' intent in dialogues with chatbot services, with only limited labeled data; (2) handling heterogeneous tasks for the better generalization ability exactly requires reliable knowledge organization, which highly relies on disentangling underlying task relations that are ignored by prior works. Motivated by those, we study a new FSTC setting where tasks come from a diverse range of data sources with possibly different data distributions as shown in Figure 1. To embrace the skills learned from multiple task sources to improve the generalization ability, we propose a novel meta-learning framework named Self-Supervised Hierarchical Task Clustering (SS-HTC), which groups tasks into different clusters based on inherent task relations in multiple levels. When a new task arrives, it can quickly take advantage of the historical knowledge learned within the cluster it belongs to.

Specifically, learning a superior task embedding is the cornerstone to disentangle underlying relations among tasks and group them into different clusters. However, a FSL task is hard to represent, as labeled training data in each FSL task are insufficient. To tackle this issue, we propose a label-oriented masked language modeling to recover the corresponding label texts using each training sample itself from the task. Such a self-supervised manner, considering informative label text semantics, encourages the model to generate more discriminative task embedding to discover reasonable task relationships even with limited label information.

After that, each task embedding is passed to a hierarchical task tree to dynamically perform soft task clustering in multiple levels, so that the knowledge is shared among highly related tasks in the same cluster but differentiated between different clusters of tasks. Then, the updated task embedding outputted by the task tree encodes the representation of the cluster it belongs to. This cluster representation is finally passed to modulate the prior knowledge, a metric space for finding nearest prototypes following (Snell et al., 2017), to be cluster-specific. In a nutshell, SS-HTC not only quickly accesses the most relevant cluster and tailors the prior knowledge to address the challenge of task heterogeneity, but also increases the model interpretability by disentangling task correlations. Empirically, extensive experiments on five public FSTC benchmark datasets demonstrate that SS-HTC significantly and consistently outperforms state-of-the-art FSTC methods by a large margin.

Our contributions can be summarized as follows. (1) A more realistic FSTC setting that allows diverse tasks with different distributions is investigated; (2) A novel SS-HTC framework is proposed to both tackle task heterogeneity and improve the interpretability by hierarchical task clustering; (3) Extensive experiments verify the effectiveness of the proposed SS-HTC method.

2 Preliminaries

Few-shot learning (FSL) Considering a task $\mathcal{T} = \{\mathcal{S}, \mathcal{Q}\}$ that contains the training set $\mathcal{S}$ and the testing set $\mathcal{Q}$, the objective of FSL is to learn a model G for this task given only a few labeled samples in $\mathcal{S}$. Typically, FSL is characterized as a N-way K-shot problem with $\mathcal{S}$ containing K labeled examples per class for N classes, i.e., $\mathcal{S} = \{(\mathbf{x}_i^j, y_i)\}_{i,j=1}^{N,K}$, where $\mathbf{x}_i^j$ is the j-th sample for the i-th class y_i. Then $\mathbf{x}^{\mathcal{Q}}$ denotes an unlabeled sample of $\mathcal{Q}$ belonging to one of the N classes and $\hat{y}^{\mathcal{Q}}$ denotes the estimated label of a model, i.e., $G(\mathcal{S}, \mathbf{x}^{\mathcal{Q}}) \rightarrow \hat{y}^{\mathcal{Q}}$. In many existing works on FSL, $\mathcal{S}$ and $\mathcal{Q}$ are also known as the *support set* and *query set*, respectively.

Traditional deep learning models would severely overfit on FSL tasks since only a few labeled samples cannot accurately represent the true data distribution, which will result in learning classifiers with high variance and generalizing poorly to new data. In order to solve the overfitting problem in FSL, Vinyals et al. (2016) proposed an effective episodic meta-learning strategy that learns a generic classifier from diverse few-shot classification tasks and then employs the classifier to a new task. The purpose of episodic training is to mimic the real testing environment where tasks contain insufficient support sets and unlabeled query sets. The consistency between training and testing environments alleviates the shift gap and boosts the generalization. Specifically, using the episodic strategy, the whole process of meta-learning can be divided into three parts: **meta-training** with training tasks $\{\mathcal{T}_{\text{train}}^k\}_{k=1}^{N_{\text{train}}} = \{\mathcal{S}^k, \mathcal{Q}^k\}_{k=1}^{N_{\text{train}}}$, **meta-validation** with validation tasks $\{\mathcal{T}_{\text{val}}^k\}_{k=1}^{N_{\text{val}}} = \{\mathcal{S}^k, \mathcal{Q}^k\}_{k=1}^{N_{\text{val}}}$, and **meta-testing** with testing tasks $\{\mathcal{T}_{\text{test}}^k\}_{k=1}^{N_{\text{test}}} = \{\mathcal{S}^k, \mathcal{Q}^k\}_{k=1}^{N_{\text{test}}}$. Note that for meta-training and meta-validation tasks, the label for the query set

is available to train the model G and to select best hyper-parameters, respectively. In this way, meta-learning algorithms are capable of adapting to new tasks effectively even with a shortage of training data for each new task.

2.1 Problem Formulation

Multi-source FSTC Prior works assume that all the tasks are sampled from a single dataset D, making the tasks lying in the same distribution. In this way, we usually split D into three parts: D_{train}, D_{dev}, and D_{test} in terms of class splits. Each part has a specific label space and disjoints with other parts. For each training episode, we first sample a label set C with N classes from D_{train}, and then use C to sample a task $\mathcal{T}^k_{\text{train}}$ containing the support set $\mathcal{S}$ and the query set $\mathcal{Q}$. Finally, we feed $\mathcal{S}$ and $\mathcal{Q}$ to the model and minimize the loss.

This assumption restricts the task diversity and degrades the model's out-of-distribution generalization. To resolve it, we assume that the tasks can be sampled from M diverse datasets $\{D^1, \ldots, D^M\}$ with possibly different distributions. For each dataset D^m, we use the same strategy to split D^m into the training, validation, and testing parts. And we sample the meta-training, meta-validation, and meta-testing tasks based on the corresponding parts of all the datasets. That is, we sample $\{\mathcal{T}_{\text{train}}\}$ from $\{D^1_{\text{train}} \cup D^2_{\text{train}} \cup \ldots D^M_{\text{train}}\}$, while each task is sampled to consist of only classes from a single dataset.

3 Method

SS-HTC aims to handle the task heterogeneity by automatically organizing tasks into a hierarchical task structure that explicitly tailors the transferable knowledge to different task clusters.The overall framework of SS-HTC is illustrated in Figure 2.

SS-HTC mainly consists of three components:

- **Prototypical network (ProtoNet)** (Snell et al., 2017) is an advanced metric-based model, which learns to predict by comparing the distance between the labeled support and unlabeled query sets. We choose it as the building block (**Base Model**) since it is computationally efficient and simple. More importantly, our framework is general and can be easily compatible with any other metric-based models, e.g., Matching Network (Vinyals et al., 2016) and Signature (Bao et al., 2020).

- **Label-Oriented Mask Language Modeling (LOMLM)** is a self-supervised learning objective to automatically learn the task embedding of each few-shot task $\mathcal{T}$ by considering informative label text semantics. LOMLM encourages the model to generate discriminative task embeddings, which are the prerequisite to identify underlying task relationships for knowledge organization and reuse.

- **Hierarchical Task Clustering (HTC)** can automatically group task knowledge into a hierarchical clustering tree, by softly assigning highly correlated tasks into the same cluster, while keeping irrelevant tasks apart. When a new task arrives, it can leverage the historical knowledge within the clusters it belongs to customize a cluster-specific metric for the prototypical network.

3.1 Prototypical Network

The prototypical network (Snell et al., 2017) is a simple yet effective metric-based method that learns to predict the label of a query sample $\mathbf{x}^{\mathcal{Q}}$ by comparing its distance with each class prototype vector. Specifically, given a N-way K-shot task $\mathcal{T}$ defined in Section 2, we use a prototype vector $\mathbf{p}_i$ as the representative feature of each class y_i, where $\mathbf{p}_i$ is the average of all the embedded support samples $\{\mathbf{x}^j_i\}^K_{j=1}$ that belong to class y_i, i.e., $\mathbf{p}_i = \frac{1}{K}\sum_{j=1}^{K} f_{\boldsymbol{\theta}}(\mathbf{x}^j_i)$, where $f_{\boldsymbol{\theta}}(\mathbf{x})$ denotes the embedding of a sample. Here, we use the pre-trained language model BERT (Devlin et al., 2019) as the powerful encoder $f_{\boldsymbol{\theta}}$. Then the probability distribution over the N classes for the query sample $\mathbf{x}^{\mathcal{Q}}$ can be calculated via a softmax function over distances between all the prototype vectors and the embedding for $\mathbf{x}^{\mathcal{Q}}$ as

$$\hat{y}^{\mathcal{Q}} = \frac{\exp(-d(f_{\boldsymbol{\theta}}(\mathbf{x}^{\mathcal{Q}}), \mathbf{p}_i))}{\sum_{i'=1}^{N} \exp(-d(f_{\boldsymbol{\theta}}(\mathbf{x}^{\mathcal{Q}}), \mathbf{p}_{i'}))}, \quad (1)$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance. The training objective is to minimize the N-way cross-entropy loss ℓ for all the query samples in the query set $\mathcal{Q}$ for each meta-training task as

$$\mathcal{L}_{\text{cls}} = \sum_{\mathcal{Q}} \ell(y^{\mathcal{Q}}, \hat{y}^{\mathcal{Q}}).$$

However, the prototypical network relies on a globally shared metric $(d, f_{\boldsymbol{\theta}})$, which may lack the ability to handle heterogeneous tasks lying in different distributions. Thus, the proposed SS-HTC method uses it as the base model and aims to improve it with the cluster-specific metric to tackle the task heterogeneity problem.

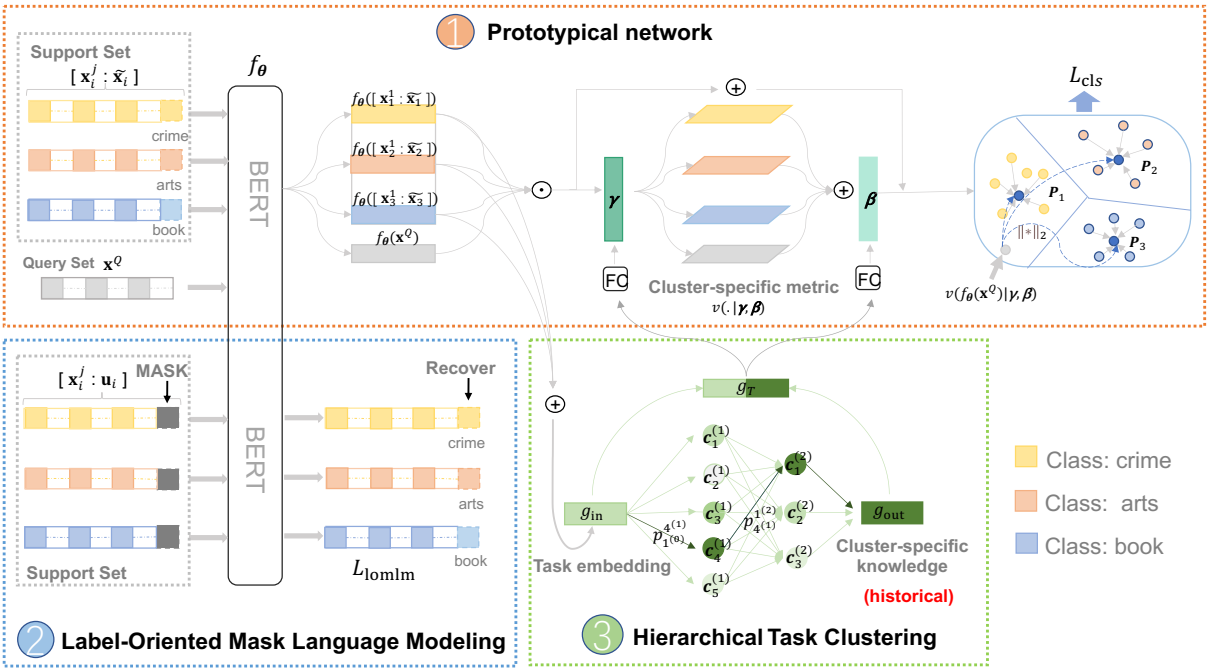


Figure 2: The SS-HTC framework with a 3-way 1-shot classification task (i.e., crime, arts, book).

3.2 Label-Oriented Mask Language Modeling

Learning a superior task embedding is the prerequisite to capture underlying correlations between tasks. On the one hand, prior works rely on learning intermediate hidden representations and then aggregate hidden representations of all the training samples as the task embedding (Zamir et al., 2018). This may be infeasible in the few-shot regime since the labeled training data $\mathcal{S}$ (i.e., $N \times K$ examples) of each few-shot task $\mathcal{T}$ are insufficient. On the other hand, existing methods for FSTC (Gao et al., 2019a; Geng et al., 2019; Sun et al., 2019; Bao et al., 2020) only treat each task as a simple N-way classification by mapping informative class label names of each task into indices $\{0, 1, ..., N-1\}$. As such, the model can only focus on discriminating among classes instead of realizing what categories to be classified. Thus, each task is actually underrepresented due to ignoring label semantics.

Inspired by those, we propose a Label-Oriented Mask Language Modeling (LOMLM) that exploits underused label semantics to enhance the task representation learning. The LOMLM uses the same denoising auto-encoding (Devlin et al., 2019) from BERT as the self-supervised learning objective. Specifically, we augment each support text sample $\mathbf{x}_i^j$ with the label name tokens $\widetilde{\mathbf{x}}_i$ of its corresponding class y_i (e.g., $\widetilde{\mathbf{x}}_i$ is "*musical instruments*" for the class $y_i = 0$). We denote the augmented support sample as $[\mathbf{x}_i^j; \widetilde{\mathbf{x}}_i]$. Then we mask each token in the label name with a special symbol [MASK], and use the remaining tokens to recover them.[1]

Let the masked tokens $\{[\text{MASK}]_t\}_{t=1}^{T=|\widetilde{\mathbf{x}}_i|}$ be $\mathbf{u}_i$. Then, the training objective of LOMLM is to reconstruct $[\mathbf{x}_i^j; \widetilde{\mathbf{x}}_i]$ from $[\mathbf{x}_i^j; \mathbf{u}_i]$ over all support samples by minimizing $\mathcal{L}_{\text{lomlm}}$, which is formulated as

$$\mathcal{L}_{\text{lomlm}} = -\sum_{i=1}^{N}\sum_{j=1}^{K} \log P([\mathbf{x}_i^j; \widetilde{\mathbf{x}}_i] | [\mathbf{x}_i^j; \mathbf{u}_i]).$$

Under the LOMLM supervision, we simply use an average pooling to aggregate the embeddings of all augmented support samples $[\mathbf{x}_i^j; \widetilde{\mathbf{x}}_i]$ as the representation of the task $\mathcal{T}$ by

$$\mathbf{g}_{\text{in}} = \text{Pool}_{i,j=1}^{N,K}(f_{\boldsymbol{\theta}}([\mathbf{x}_i^j; \widetilde{\mathbf{x}}_i])). \quad (2)$$

3.3 Hierarchical Task Clustering

To cluster tasks into different groups, where the knowledge from similar historical tasks can be accumulated together and transferred to newly related tasks, we propose a hierarchical task clustering (HTC) to dynamically locate which cluster the task belongs to. The hierarchical structure adopts the top-down hierarchy design to imitate the product taxonomy from coarse to fine granularity (e.g., the "*electronics*" category has more specific subcategories such as "*laptop*", "*phone*", and "*TV*"). Given complex dependencies among tasks, hierarchical levels of task clusters are more sufficient to

[1] We explain no information leakage in Appendix A.1.

capture real-world task relations than the flat clustering (Kim and Xing, 2010). This allows the task organization and reuse in a coarse-to-fine manner, which can better disentangle inherent task relations such that transferable knowledge among tasks can be maximally leveraged.

In the hierarchical cluster tree, each task $\mathcal{T}$ is soft-assigned into the clusters in each level to encourage less information loss compared with the hard assignment and allow SS-HTC to be trained in an end-to-end manner. Specifically, the assignment score for the next level is a function of the task embedding at the current level. For example, we assign the task embedding $\mathbf{g}_o^{(l)}$ in the o-th cluster of the (l)-th level to the o'-th cluster of the $(l+1)$-th level with the probability $p_{o^{(l)}}^{o'^{(l+1)}}$, which is computed by applying the softmax function over Euclidean distances between $\mathbf{g}_o^{(l)}$ and all the $(l+1)$-th level cluster centers $\{\mathbf{c}_{o'}^{(l+1)}\}_{o'=1}^{O^{(l+1)}}$

$$p_{o^{(l)}}^{o'^{(l+1)}} = \frac{\exp(-||\mathbf{g}_o^{(l)} - \mathbf{c}_{o'}^{(l+1)}||_2^2/2\sigma^2)}{\sum_{o'=1}^{O^{(l+1)}} \exp(-||\mathbf{g}_o^{(l)} - \mathbf{c}_{o'}^{(l+1)}||_2^2/2\sigma^2)},$$

where σ^2 is a scaling factor to control the distance between tasks and clusters and $O^{(l+1)}$ denotes the number of clusters in the $(l+1)$-th level. Then, the task embedding $\mathbf{g}_{o'}^{(l+1)}$ of the o'-th cluster in the $(l+1)$-th level can be calculated by the weighted sum of all the task embeddings in the previous l-th level as

$$\mathbf{g}_{o'}^{(l+1)} = \sum_{o=1}^{O^{(l)}} p_{o^{(l)}}^{o'^{(l+1)}} \tanh(\mathbf{W}_{o'}^{(l+1)} \mathbf{g}_o^{(l)} + \mathbf{b}_{o'}^{(l+1)}),$$

where $\mathbf{W}_{o'}^{(l+1)}$ and $\mathbf{b}_{o'}^{(l+1)}$ are learnable parameters. The full pipeline of HTC starts from $l=0$ and $O^{(l)} = 1$, where the initialization for $\mathbf{g}_1^{(0)}$ is the input task embedding $\mathbf{g}_{\text{in}}$ defined in Eq. (2), and ends at $O^{(L)} = 1$. The output embedding $\mathbf{g}_{\text{out}} = \mathbf{g}_1^{(L)}$ from the tree encrypts the cluster-specific historical knowledge that can be transferred to the input task. *Note that we provide more details for the working mechanism of HTC in our Appendix A.2.*

Cluster-specific feature transformation After obtaining the cluster-specific knowledge $\mathbf{g}_{\text{out}}$ from the tree that is highly correlative and transferable to the task, we concatenate the input and output task embeddings for the tree as the final task embedding, i.e., $\mathbf{g}_{\mathcal{T}} = \mathbf{g}_{\text{in}} \oplus \mathbf{g}_{\text{out}}$. The task embedding $\mathbf{g}_{\mathcal{T}}$ is used to learn the cluster-specific feature transformation $v(\cdot|\boldsymbol{\gamma}, \boldsymbol{\beta})$ for the augmented support samples and query samples, which consists of two factors $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ both derived from $\mathbf{g}_{\mathcal{T}}$ as

$$\boldsymbol{\gamma} = \rho(\mathbf{W}_\gamma \mathbf{g}_{\mathcal{T}} + \mathbf{b}_\gamma),$$
$$\boldsymbol{\beta} = \rho(\mathbf{W}_\beta \mathbf{g}_{\mathcal{T}} + \mathbf{b}_\beta),$$

where ρ denote the ReLU function. $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are learnable scaling and shift parameters of the feature-wise transformation, which can dynamically adjust feature representations to be more discriminative based on the cluster-specific task embeddings such that it can well adapt to diverse task distributions. Recall that $f_{\boldsymbol{\theta}}(\mathbf{x})$ is the BERT representation of a sample $\mathbf{x}$, where $\mathbf{x}$ can be an augmented support sample $[\mathbf{x}_i^j; \widetilde{\mathbf{x}}_i]$ or a query sample $\mathbf{x}^{\mathcal{Q}}$. For simplicity, let $\mathbf{h} = f_{\boldsymbol{\theta}}(\mathbf{x})$, then the two factors will make a residual affine transformation $v(\cdot|\boldsymbol{\gamma}, \boldsymbol{\beta})$ on $\mathbf{h}$ as

$$v(\mathbf{h}|\boldsymbol{\gamma}, \boldsymbol{\beta}) = \rho((\mathbf{1} + \boldsymbol{\gamma}) \odot \mathbf{h} + \boldsymbol{\beta}) + \mathbf{h},$$

where $\odot$ is the element-wise multiplication. With the aid of the proposed SS-HTC, we will use the cluster-specific transformed embeddings $v(f_{\boldsymbol{\theta}}(\mathbf{x})|\boldsymbol{\gamma}, \boldsymbol{\beta})$ instead of the BERT feature embedding $f_{\boldsymbol{\theta}}(\mathbf{x})$ for the inference of the prototypical network as defined in Eq. (1).

3.4 Joint Training

We combine each component loss into an overall object function as

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{lomlm}},$$

where λ is a hyper-parameter to balance the classification loss and the LOMLM loss. The goal of joint learning is to learn superior task embeddings to guide the cluster-specific discriminative learning for the ultimate few-shot text classification.

4 Experiments

4.1 Setup

Datasets We evaluate SS-HTC on five FSTC benchmark datasets: Amazon Product Review (He and McAuley, 2016), 20 Newsgroups (Lang, 1995), HuffPost (Misra, 2018), Reuters (Lewis, 1997), and RCV1 (Lewis et al., 2004). Following (Bao et al., 2020), we use the same class splits to divide each dataset into meta-training, meta-validation and meta-testing parts, from which N-way K-shot tasks are randomly sampled.

Setting In experiments, tasks can be sampled from multiple diverse datasets. Thus, all models are trained and evaluated on the combination of the five aforementioned benchmark datasets instead of each

dataset separately. Following prior works on FSTC, the classification accuracy is used as the evaluation metric as each task is under the few-shot learning setting and has no data imbalance issue. Moreover, we use the average accuracy on randomly sampled $1,000$ meta-testing tasks for each dataset as the final results to avoid the problem of randomness. All the experiments repeat 3 times and average results over 3 runs are reported.

Baselines For a fair comparison, we use the BERT as the base encoder for all baselines.

• **Supervised learning. BERT (FT)** (Chen et al., 2019) trains a BERT with a generic N-way classifier on all meta-training tasks and finetunes it on the support set and evaluate it on the query set of each meta-testing task independently.

• **Gradient-based meta-learning** methods aim to learn a well-generalized model initialization that can be adapted to new tasks within a few optimization steps. (i) **Reptile** (Nichol et al., 2018) is a fast first-order gradient approximation of MAML which could be hardly optimized based on BERT (Finn et al., 2017). (ii) **PMAML** (Zhang et al., 2019) employs the masked language model pretraining before using the first-order MAML.

• **Metric-based meta-learning** methods are to learn an invariant metric space where classes can be differentiated between each other. (i) **MatchNet** (Vinyals et al., 2016) uses an attention-based scheme where the cosine distance is used as the metric. (ii) **ProtoNet** (Snell et al., 2017) learns a metric space by minimizing the Euclidean distance between class prototype and query samples. (iii) **InductionNet** (Geng et al., 2019) encapsulates different classes by a dynamic routing induction method. (iv) **HybridAPN** (Gao et al., 2019a) is a hybrid attention prototypical network that exploits a hybrid attention mechanism. (v) **HierAPN** (Sun et al., 2019) is a hierarchical attention prototypical network that designs a hierarchical attention mechanism. (vi) **Signature** (Bao et al., 2020) utilizes the distributional statistics to implement the attention transfer between tasks. (vii) **DEM** (Ohashi et al., 2021) introduces a difference extractor to derive distinctive label representations with multi-task learning based on ProtoNet.

4.2 Implementation details

Environment Our proposed SS-HTC model and baseline methods are implemented in TensorFlow 2.4.0 with CUDA 10.1, using Python 3.7.0 from Anaconda 4.9.2. All the models are trained/tested on a single TESLA V100-PCIE 32GB GPU with Linux system.

Encoder We use the BERT-base model: *bert-base-uncased* (Wolf et al., 2019) model as the encoder, which has 12 layers, 768-dimensional hidden representations, 12 heads, and 110M parameters in total. We use the pooled representation (i.e., averaged token embeddings) as the sentence embedding since we have found that [CLS] embedding performs very poorly, even worse than CNN encoders under the few-shot setting. The BERT is jointly optimized with other parameters during the training stage.

Initialization & Training For all the experiments, SS-HTC is optimized by the Adam algorithm (Kingma and Ba, 2014) for training. The maximal sentence length is 450. The weight matrices are initialized with a uniform distribution $U(-0.01, 0.01)$. Gradients with the $l2$ norm larger than 40 are normalized to be 40. To alleviate overfitting, we perform early stopping on the meta-validation tasks.

Hyperparameter The hyper-parameters are manually tuned on the average accuracy of the 10% randomly held-out meta-training sets. The initial learning rate is 10^{-5}, which is tuned amongst $\{10^{-6}, 5\times10^{-6}, 1\times10^{-5}, 5\times10^{-5}\}$. The weight λ for $\mathcal{L}_{\text{lomlm}}$ is 0.1, which is tuned amongst {0.01, 0.03, 0.1, 0.3}. The scaling factor σ^2 is 2.0. Due to the limited GPU memory, we only feed one task to SS-HTC for each step.

4.3 Main Results

K-Shot Evaluation. We present in Table 1 experimental results in terms of different shots under the setting of five ways/classes. Based on the results, we can observe: **SS-HTC**: SS-HTC significantly and consistently outperforms all the baseline methods on five datasets by a large margin (i.e., 1-shot:+9.81%, 5-shot:+5.19% average accuracy) over the best baselines (i.e., Signature and DEM).

• **Supervised method**: Even with powerful pre-trained language models (PLMs) like BERT, the supervised method BERT (FT) still performs very poorly in the few-shot regime. This circumstance has also been shown in prior studies (Yogatama et al., 2019), which shows that PLMs highly rely on sufficient fine-tuning data for downstream tasks.

• **Gradient-based methods**: As gradient-based methods, Reptile and PMAML show inferior per-

Model	20 News		Amazon		Huffpost		Reuters		RCV1		Avg	
	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot
Supervised learning												
BERT (FT)	28.30	34.01	34.35	43.93	23.50	28.11	37.01	51.35	30.42	39.88	30.72	39.46
Gradient-based meta learning												
Reptile	33.78	40.23	39.86	55.01	25.69	36.20	47.55	64.56	40.02	56.33	37.38	50.47
PMAML	34.46	40.25	38.41	53.69	26.18	35.44	46.57	65.13	38.99	57.70	36.92	50.44
Metric-based meta learning												
MatchNet	38.03	40.42	39.26	34.02	33.16	58.02	54.27	38.16	40.61	42.44	41.07	42.61
InductionNet	41.35	43.52	46.36	43.18	38.09	42.32	69.12	66.92	46.04	50.84	48.19	49.36
ProtoNet	49.30	65.51	68.91	84.79	46.54	65.55	73.46	87.45	49.32	69.70	57.51	74.60
HybridAPN	48.56	57.80	68.92	77.25	44.39	53.19	80.21	88.42	55.95	66.52	59.61	68.64
HierAPN	52.88	57.66	66.35	75.64	41.87	51.62	80.28	92.58	54.83	61.54	59.24	67.81
Signature	52.48	66.20	66.64	84.44	45.32	63.20	83.52	93.20	53.58	69.20	60.31	75.25
DEM	49.88	57.93	54.96	73.56	52.34	69.66	87.27	95.21	58.18	76.07	60.53	74.49
SS-HTC	**58.77**†	**69.24**†	**75.92**†	**86.84**†	**63.72**†	**71.88**†	**89.36**†	**95.98**†	**63.91**†	**78.24**†	**70.34**†	**80.44**†
Δ	(+6.29)	(+3.04)	(+8.65)	(+2.40)	(+11.38)	(+2.22)	(+2.09)	(+0.77)	(+5.73)	(+2.17)	(+9.81)	(+5.19)

Table 1: Main results: 5-way K-shot evaluation. Δ refers to the improvements over the best baseline. † means the statistically significant improvement with paired sample t-test with p-value < 0.01.

formance to metric-based baselines in FSTC. This phenomenon has also been observed in recent FSTC works (Gao et al., 2019a; Bao et al., 2020). The gradient-based methods mainly focus on low-noise vision tasks, which makes them hard to directly deal with diverse and noisy text data in FSTC tasks, especially for our setting that tasks are coming from multiple resources with large diversity.

• **Metric-based methods**: (i) Compared with gradient-based methods, most metric-based baselines can generally obtain better results for FSTC. (ii) Recent proposed text-specific metric-based methods like HybridAPN and HierAPN have better performance than their base model - ProtoNet when tasks are all sampled from a single dataset (Gao et al., 2019a; Sun et al., 2019). However, when tasks are heterogeneous from diverse datasets in our setting, they do not outperform ProtoNet. This indicates that their sophisticated metric designs may not be able to handle the task heterogeneity due to the global knowledge-sharing strategies used. (iii) SS-HTC can outperform those metric-based baselines. This is because that SS-HTC can customize the transferable knowledge to be cluster-specific and preserve knowledge generalization among highly related tasks by taking advantage of the dynamic task clustering.

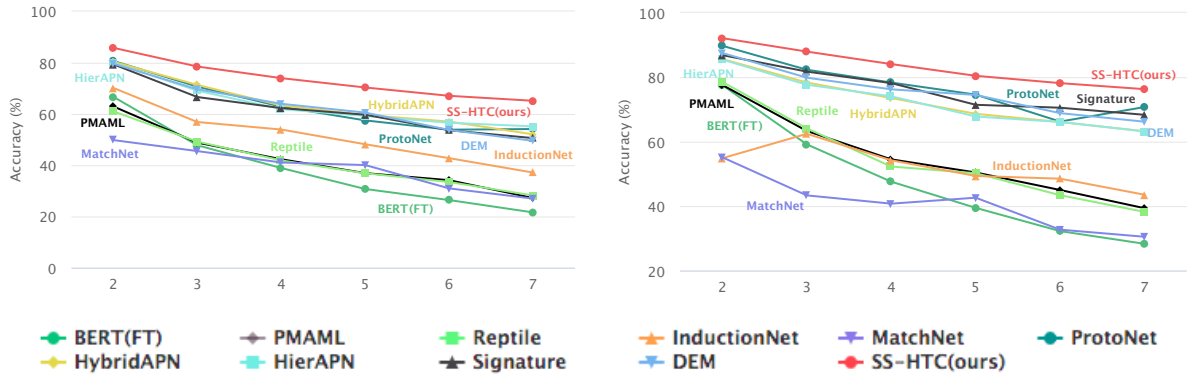
N-way Evaluation. We present in Figure 3 the results in terms of different ways with a fixed number of shots. We report the average accuracy across all the five datasets with $N = 2, 3, ..., 7$. Generally, as the number of ways increases, the performance degrades as the FSTC tasks become more difficult. We can observe that SS-HTC performs better than other baselines and that the gap among them becomes larger as the number of ways increases. This indicates the proposed SS-HTC method is less sensitive to the difficulty of the FSTC task by leveraging the knowledge from the most similar tasks based on hierarchical task clustering.

4.4 Ablation Study

To verify the efficacy of each component, we progressively incorporate the hierarchical task clustering (HTC) and label-oriented masked language modeling (LOMLM) into the base model (i.e., ProtoNet). We present the ablation results in Table 2.

• **w/ HTC** v.s. **w/o HTC**: For ProtoNet+HTC, we use the proposed HTC method to dynamically organize tasks into hierarchical clusters, where the knowledge from similar tasks can be accumulated together. As such, each new incoming task can leverage the transferable knowledge within the cluster it belongs to and customize the cluster-specific metric for few-shot learning. We observe HTC can bring a significant gain (i.e., 2.45%) over ProtoNet in terms of the average accuracy. This shows the effectiveness of HTC to handle heterogeneous tasks lying in different distributions.

• **w/ LOMLM** v.s. **w/o LOMLM**: For ProtoNet+HTC, we simply average embeddings of all support samples and their corresponding label texts as the embedding of a task without any supervision. Thus, this task embedding could be underrepresented. By incorporating the LOMLM, the task embedding is enhanced to be more label-aware to discriminate among classes. According to Table 2, we observe that adding LOMLM can achieve an additional 6.88% gain in terms of the average accuracy, which is a very large improvement over ProtoNet+HTC. This implies that a superior task embedding is critical to better disentangling task relations and customize the cluster-specific metric.



(a) **N-way 1-shot** (b) **N-way 5-shot**

Figure 3: Main results: Average performance for N-way evaluation with the fixed numbers of shots.

Model	1-shot	5-shot	Avg	Gain
ProtoNet	57.51	74.60	66.06	-
ProtoNet+HTC	60.98	76.04	68.51	+ 2.45
ProtoNet+LOMLM+HTC - a.k.a SS-HTC	**70.34**	**80.44**	**75.39**	+6.88

Table 2: Average results for 5-way classification.

4.5 The Effect of Tree Structure

We further study the effect of tree structure to the performance. We vary the tree structure and record the results in Table 3. From the results, we can observe that the proposed hierarchical clustering shows the superiority over the flat task clustering. For the hierarchical clustering, we can see that too few clusters may be insufficient to learn the task clustering characteristic (e.g., the case (2,2,1)). When we increase the number of clusters, SS-HTC can achieve better results (e.g., case (5,3,1)) until reaching a stable status (e.g., case (5,5,1)). This indicates that more clusters introduce more parameters and may result in the overfitting problem.

Num. of Clu.	1-shot	5-shot	Avg
Flat clustering			
(5,1)	68.18	79.12	73.65
(15,1)	68.41	79.41	73.91
Hierarchical clustering			
(2,2,1)	68.14	79.20	73.67
(3,2,1)	68.03	79.51	73.77
(5,3,1)	**70.34**	**80.44**	**75.39**
(5,4,1)	70.12	80.18	75.15
(5,5,1)	70.30	80.38	75.34

Table 3: Comparison among different cluster #. $(\cdot,\cdot,\cdot)$ denotes the cluster # from the bottom to the top layer. Average accuracy for 5-way classification is reported.

4.6 Visualization of Hierarchical Task Tree

To demonstrate that the proposed SS-HTC method can automatically disentangle the underlying task relationship, we visualize the SS-HTC with cluster structure $(5, 3, 1)$ for tasks from each dataset. Specifically, we first select 1,000 5-way 1-shot tasks randomly from each dataset and show their averaged soft-assignments of clusters (C1, C2, C3, C4, C5) in the first layer. As illustrated in the left subfigure in Figure 4 where a darker color means a higher probability, we can see that different datasets mainly activate different clusters: Reuters→C2, Amazon→C3, RCV1→C4, and 20News→C1. Particularly, Huffpost activates both C2 and C4, which indicates that the Huffpost and Reuters datasets may have a large overlap. By checking the classes sets of both datasets, we have found that several classes in the two datasets are highly-related (e.g., Huffpost: "*taste*" and "*word news*", Reuters: "*sugar*" and "*wholesale price Iindex*").

Besides, we also explore the activated task clusters (A, B, C) in the second layer which further accumulates the transferable knowledge among tasks from different datasets. We observe tasks from different datasets that have similar classes are highly aggregated into the same cluster. Meanwhile, tasks from the same dataset that contains different classes can activate different clusters. For example, in the #1 case of Figure 4 Right, a 5-way task from **20News** with the class set {"*alt atheism*", "*soc religion christian*", "*talk politics guns*", "*talk politics misc*", "*talk religion misc*"} and a 5-way task from **RCV1** with the class set "{"*religion*", "*equity markets*", "*domestic politics*", "*interbank markets*", "*money markets*"}" both activate the first cluster A, since the two tasks are all related to religion and politics. Similarly, in the #2 case, a 5-way task from **20News** with the class set {"*books*", "*clothing shoes jewelry*", "*electronics*", "*musical instruments*", "*tools home improvement*"} and a 5-way task from **Huffpost** with the

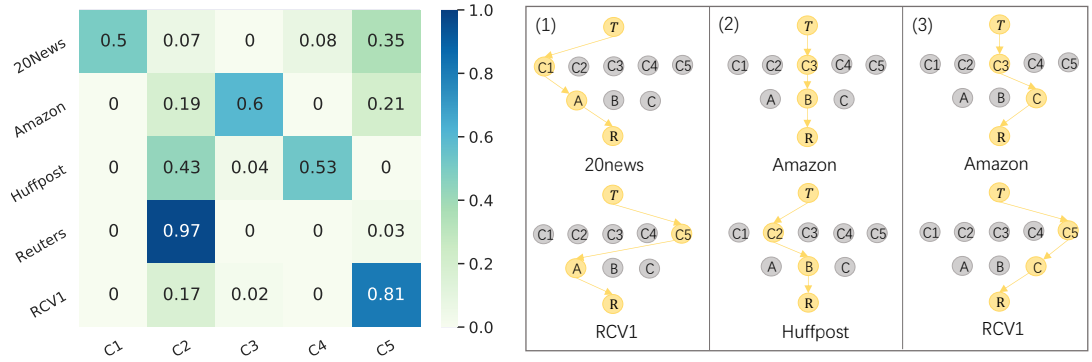


Figure 4: Left: visualization of average soft-assignment $p^{o'^{(l+1)}}_{o^{(l)}}$ of 1000 random tasks for each dataset. Right: hierarchical structure learned from different tasks. The most activated cluster is marked in yellow.

class set {"*arts culture*", "*good news*", "*environment*", "*tech*", "*style*"} both activate the second cluster B because they are all about culture and life-related stuffs. In the #3 case, a 5-way task from **Amazon** with the class set {"*Books*", "*kindle store*", "*movies tv*", "*office products*", "*tools home improvement*""} and a 5-way task from **RCV1** with the class set {"*economics*", "*government finance*", "*management*", "*performance*", "*share listings*"} are both assigned to the third cluster C as they both concern the economy and education. Those qualitative results indicate that the proposed SS-HTC model can capture the latent relations between diverse tasks to improve the model interpretability.

5 Related Work

Meta-Learning Inspired by human beings' ability to transfer knowledge from previous experiences (Pan and Yang, 2009; Li et al., 2017, 2018, 2019b,a), meta-learning (Vinyals et al., 2016; Finn et al., 2017) has become the mainstream paradigm to resolve few-shot learning problems. Prior studies mainly focus on low-noise vision tasks (Snell et al., 2017; Sung et al., 2018; Nichol et al., 2018; Oreshkin et al., 2018; Liu et al., 2019). Recently, those techniques have been initiated to low-resource NLP problems such as few-shot text classification (Yu et al., 2018; Wu et al., 2019; Geng et al., 2019; Sun et al., 2019; Geng et al., 2020; Bao et al., 2020; Wang et al., 2021; Ohashi et al., 2021), relation classification (Han et al., 2018; Gao et al., 2019a; Obamuyide and Vlachos, 2019; Gao et al., 2019b), machine translation (Gu et al., 2018), knowledge graph completion (Huang et al., 2022; Wang et al., 2022), and natural language understanding (Dou et al., 2019; Bansal et al., 2020a,b; Li et al., 2021) with minimal supervision. Different from them that globally share the prior knowledge across homogeneous tasks within a single source, SS-HTC can embrace the skills learned from multiple heterogeneous sources to improve the out-of-distribution robustness More importantly, they neglect underlying task relations in the low-data regime, which is imperative to automatically organize knowledge from heterogeneous tasks.

Label-aware Modeling To alleviate data scarcity, label-aware methods are recently investigated (Yin et al., 2019; Puri and Catanzaro, 2019; Meng et al., 2020; Halder et al., 2020) to incorporate label semantics into text representation learning. Yin et al. (Yin et al., 2019) incorporate label texts into text samples and convert the text classification into a text entailment task. Yu et al. (Meng et al., 2020) propose the category vocabulary, which can be good label supplements for our LOMLM to enrich the label semantics in the future work. However, those methods are less investigated in task adaptation for FSTC. More importantly, our ultimate goal aims to leverage label information to enhance few-shot task representations for discovering and disentangling inherent and complicated task correlations. This can facilitate the knowledge organization and handle heterogeneous new tasks as well as improving the model interpretability.

6 Conclusion

In this paper, we propose the self-supervised hierarchical task clustering (SS-HTC) method to tackle the task heterogeneity for FSTC by dynamically organizing the tasks into hierarchical clusters and customize the cluster-specific knowledge. Extensive experiments on various FSTC benchmark datasets quantitatively and qualitatively demonstrate the effectiveness of SS-HTC. In the future, the proposed SS-HTC can be potentially generalized to the multilingual few-shot setting (Hu et al., 2020).

7 Limitations

Although we introduce a more realistic and practical problem setting for few-shot learning and verify the proposed SS-HTC method on extensive experiments, there are still some future directions that need further investigation and exploration. Firstly, our proposed setting is supposed to generalize to more heterogeneous NLP tasks under the few-shot regime instead of restricting to text classification. Secondly, how to dynamically adapt the task organization structures like humans in terms of input tasks is still underexploited. We view our works as the start point and will further explore those interesting problems in the future work.

Acknowledgements

This work is supported by NSFC general grant under grant no. 62076118 and Shenzhen fundamental research program JCYJ20210324105000003.

References

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2020a. Learning to few-shot learn across diverse natural language classification tasks. In *COLING*, pages 5108–5123.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020b. Self-supervised meta-learning for few-shot natural language classification tasks. In *EMNLP*, pages 522–534.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *ICLR*.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *ICLR*.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *WSDM*, pages 151–159.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *EMNLP-IJCNLP*, pages 1192–1197.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019a. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *AAAI*, volume 33, pages 6407–6414.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019b. FewRel 2.0: Towards more challenging few-shot relation classification. In *EMNLP-IJCNLP*, pages 6250–6255.

Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. In *ACL*, pages 1087–1094.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *EMNLP-IJCNLP*, pages 3904–3913.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *EMNLP*, pages 3622–3631.

Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. 2020. Task-aware representation of sentences for generic text classification. In *COLING*, pages 3202–3213.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *EMNLP*, pages 4803–4809.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *ICML*, volume 119, pages 4411–4421.

Zijie Huang, Zheng Li, Haoming Jiang, Tianyu Cao, Hanqing Lu, Bing Yin, Karthik Subbian, Yizhou Sun, and Wei Wang. 2022. Multilingual knowledge graph completion with self-supervised adaptive graph alignment. In *ACL*.

Seyoung Kim and Eric P Xing. 2010. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

David Lewis. 1997. Reuters-21578 text categorization test collection, distribution 1.0. *http://www.research/.att.com*.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *JMLR*, 5(Apr):361–397.

Zheng Li, Xin Li, Ying Wei, Lidong Bing, Yu Zhang, and Qiang Yang. 2019a. Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning. In *EMNLP*, pages 4590–4600, Hong Kong, China.

Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. 2018. Hierarchical attention transfer network for cross-domain sentiment classification. In *AAAI*, pages 5852–5859.

Zheng Li, Ying Wei, Yu Zhang, Xiang Zhang, and Xin Li. 2019b. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In *AAAI*, pages 4253–4260.

Zheng Li, Danqing Zhang, Tianyu Cao, Ying Wei, Yiwei Song, and Bing Yin. 2021. Metats: Meta teacher-student network for multilingual sequence labeling with minimal supervision. In *EMNLP*, pages 3183–3196.

Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *IJCAI*, pages 2237–2243.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sungju Hwang, and Yi Yang. 2019. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *EMNLP*, pages 9006–9017.

Rishabh Misra. 2018. News category dataset.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Abiola Obamuyide and Andreas Vlachos. 2019. Model-agnostic meta-learning for relation classification with limited supervision. In *ACL*, pages 5873–5879.

Sora Ohashi, Junya Takayama, Tomoyuki Kajiwara, and Yuki Arase. 2021. Distinct label representations for few-shot text classification. In *ACL-IJCNLP*, pages 831–836.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, pages 721–731.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *TKDE*, 22(10):1345–1359.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087.

Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical attention prototypical networks for few-shot text classification. In *EMNLP-IJCNLP*, pages 476–485.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *NeurIPS*, pages 3637–3645.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. 2018. Toward multimodal model-agnostic meta-learning. *arXiv preprint arXiv:1812.07172*.

Jixuan Wang, Kuan-Chieh Wang, Frank Rudzicz, and Michael Brudno. 2021. Grad2task: Improved few-shot text classification using gradients for task representation. In *NeurIPS*.

Ruijie Wang, Zheng Li, Dachun Sun, Shengzhong Liu, Jinning Li, Bing Yin, and Tarek Abdelzaher. 2022. Learning to sample and aggregate: Few-shot reasoning over temporal knowledge graphs.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jiawei Wu, Wenhan Xiong, and William Yang Wang. 2019. Learning to learn and predict: A meta-learning approach for multi-label classification. In *EMNLP-IJCNLP*, pages 4354–4364.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *EMNLP*, pages 3914–3923.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *NAACL*, pages 1206–1215.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. In *AAAI*.

N. Zhang, Z. Sun, S. Deng, J. Chen, and H. Chen. 2019. Improving few-shot text classification via pretrained language representations.

A Clarification

In this section, we give more clarification regarding the details of problem setting, framework as well as hierarchical task clustering.

The proposed SS-HTC framework exactly comes from the collective power of self-supervised LOMLM and HTC. By this means, SS-HTC balances between globally shared meta-knowledge and cluster-specific meta-knowledge, where the transferable knowledge can be adapted to different clusters of tasks, while it is still shared among highly correlated tasks within the same cluster.

A.1 No Information leakage claim

There is no information leakage for the proposed Label-Oriented Mask Language Modeling (LOMLM) module. We only utilize masked label tokens for support samples (training set) instead of query samples (testing set) in each task. For meta-learning, the final performance evaluation is based on query samples of each meta-testing task that has disjoint classes with all meta-training tasks.

A.2 Hierarchical Task Clustering

The characteristics of HTC can be summarized as two aspects: (1) the hierarchical task clusters $\{\mathbf{c}_o^{(l)}\}_{o=1}^{O^{(l)}}$ in each (l)-th level of the tree are learnable and randomly initialized, which are shared by all tasks. We only need to feed each task embedding into the tree, automatically obtain the soft assignment to each cluster, and output the cluster-specific historical knowledge used for the prototypical network. The structure of the hierarchical tree is predefined since we found that jointly learning with additional structures can bring more challenges into the optimization. Despite that, the cluster representations and their connection weights are jointly learned with other parameters in an online manner to model complex task relationships; (2) hierarchical clustering tree is optimized on the task level instead of the class level, which can capture more enriched task-specific information beyond the class itself. We found this information is particularly useful to handle the diversity of few-shot tasks, as our practical setting allows tasks to be sampled from a diverse range of data sources with possibly different data distributions.

B Baselines

We provide the available open source code for the baseline methods we compare with, including:

- **Supervised learning**.
  - **BERT (FT)** (Chen et al., 2019)[2]
- **Gradient-based meta-learning**
  - **Reptile** (Nichol et al., 2018)[3]
  - **PMAML** (Zhang et al., 2019)[4]
- **Metric-based meta-learning**
  - **MatchNet** (Vinyals et al., 2016)[5]
  - **ProtoNet** (Snell et al., 2017)[6]
  - **InductionNet** (Geng et al., 2019)[7]
  - **HybridAPN** (Gao et al., 2019a)[8]
  - **Signature** (Bao et al., 2020)[9]
  - **DEM** (Ohashi et al., 2021)[10]

For **HierAPN** (Sun et al., 2019), we reimplement it according to the original paper since the source code is not publicly available.

[2] https://github.com/wyharveychen/CloserLookFewShot
[3] https://github.com/openai/supervised-reptile
[4] https://github.com/zxlzr/FewShotNLP
[5] https://github.com/gitabcworld/MatchingNetworks
[6] https://github.com/jakesnell/prototypical-networks
[7] https://github.com/YujiaBao/Distributional-Signatures
[8] https://github.com/thunlp/HATT-Proto
[9] https://github.com/YujiaBao/Distributional-Signatures
[10] https://github.com/21335732529sky/difference_extractor