MetaSLRCL: A Self-Adaptive Learning Rate and Curriculum Learning Based Framework for Few-Shot Text Classification

Kailin Zhao, Xiaolong Jin*, Saiping Guan, Jiafeng Guo, Xueqi Cheng
CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences;
School of Computer Science and Technology, University of Chinese Academy of Sciences
{zhaokailin17z, jinxiaolong, guansaiping}@ict.ac.cn
{guojiafeng, cxq}@ict.ac.cn

Abstract

Due to the lack of labeled data in many realistic scenarios, a number of few-shot learning methods for text classification have been proposed, among which the meta learning based ones have recently attracted much attention. Such methods usually consist of a learner as the classifier and a meta learner for specializing the learner to different tasks. For the learner, learning rate is crucial to its performance. However, existing methods treat it as a hyper parameter and adjust it manually, which is time-consuming and laborious. Intuitively, for different tasks and neural network layers, the learning rates should be different and self-adaptive. For the meta learner, it requires a good generalization ability so as to quickly adapt to new tasks. Motivated by these issues, we propose a novel meta learning framework, called MetaSLRCL, for few-shot text classification. Specifically, we present a novel meta learning mechanism to obtain different learning rates for different tasks and neural network layers so as to enable the learner to quickly adapt to new training data. Moreover, we propose a task-oriented curriculum learning mechanism to help the meta learner achieve a better generalization ability by learning from different tasks with increasing difficulties. Extensive experiments on three benchmark datasets demonstrate the effectiveness of MetaSLRCL.

1 Introduction

Text classification is one of the most concerned tasks in Natural Language Processing (NLP). At present, most text classification methods are based on supervised learning with a large amount of labeled data. But there is not so much labeled data, even source data, in many scenarios (e.g., news classification in specific domains). Some distant supervision methods (Mintz et al., 2009) have thus been proposed to handle this problem. However, this kind of methods may add a large proportion of noisy training data (Zeng et al., 2014). Because of this, it is a big challenge for traditional supervised learning methods to work well in the scenarios with very limited training data. As a result, few-shot text classification has attracted much attention in recent years, where there are only a few labeled instances available for each class.

The concept of few-shot learning was formally put forward by (Li et al., 2003). They presented a method for learning from classes with few data, by incorporating generic knowledge which may be obtained from previously learned models of unrelated classes. The existing few-shot learning methods can be divided into three categories (Gao et al., 2019), namely, model fine-tuning based (e.g., (Howard and Ruder, 2018; Nakamura and Harada, 2019)), metric learning based (e.g., (Snell et al., 2017; Vinyals et al., 2016)), and meta learning based methods (e.g., (Finn et al., 2017; Munkhdalai and Yu, 2017)). In recent years, meta learning based methods have attracted lots of interests. However, they still suffer from some challenges.

A meta learning method is composed of a learner and a meta learner. For the learner, learning rate is crucial to its performance. Nevertheless, in existing methods, it is treated as a hyper parameter and needs to be adjusted manually, which is time-consuming and laborious. Intuitively, for different tasks and different neural network layers, their learning rates should be different. On the other hand, the present meta learning methods cannot be quickly generalized to new tasks (Zheng et al., 2021) and a good generalization ability to new tasks is necessary for the meta learner. And curriculum learning can help models obtain better generalization performance by guiding the training process towards better regions in the parameter space, i.e., into local minima of the descent procedure associated with good generalization (Bengio et al., 2009).

*Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2065–2074
October 12–17, 2022.

For the above reasons, we propose a novel meta learning framework, called MetaSLRCL, for few-shot text classification, which contains two main mechanisms, i.e., Self-adaptive Learning Rates for the learner and a task-oriented Curriculum Learning mechanism for the meta learner. Our general contributions are three-fold. 1) We present a novel meta learning mechanism with self-adaptive learning rates, which enables different tasks and neural network layers to obtain different learning rates; 2) We introduce curriculum learning for the first time, to the best of our knowledge, into few-shot learning. Unlike traditional instance-oriented curriculum learning, the proposed task-oriented curriculum learning mechanism gradually learns from different tasks with increasing difficulties; 3) MetaSLRCL is evaluated with three typical types of text classification, i.e., relation classification, news classification and topic classification, on three benchmark datasets, namely, FewRel80, 20Newsgroup and DBPedia Ontology, respectively. Experimental results demonstrate its superior performance on all datasets.

2 Related Works

2.1 Few-shot Learning

Few-shot learning is to learn how to solve problems from few data. As aforesaid, the existing mainstream methods can be divided into three categories. The model fine-tuning based mbethods learn how to fine-tune general-purpose models to specialized tasks (Howard and Ruder, 2018; Nakamura and Harada, 2019). The metric learning based methods learn a semantic embedding space upon a distance function (Snell et al., 2017; Vinyals et al., 2016). The meta learning based methods learn a learning strategy to make them well adapt to new tasks (Finn et al., 2017; Munkhdalai and Yu, 2017). Furthermore, according to the different kinds of meta knowledge the meta learner learns, the meta learning based methods can be further divided into three sub-categories, i.e., initial parameter (Finn et al., 2017; Raghu et al., 2019; Jamal and Qi, 2019), hyper parameter (Wu et al., 2019) and optimizer based methods (Santoro et al., 2016; Munkhdalai and Yu, 2017). The initial parameter based methods learn parameter initialization for fast adaptation; The hyper parameter based methods learn a good hyper parameter setting for the learner; And, the optimizer based methods learn a meta-policy to update the parameters of the learner. Some methods of the hyper parameter based category in Computer Vision (CV) (e.g., MAML++ (Antoniou et al., 2019) and ALFA (Baik et al., 2020)) have explored to learn the learning rate. However, these methods usually consider from a single perspective, e.g., the network layer or loop perspective. Specifically, MAML++ learns the learning rate from the network layer perspective, while ALFA learns it from the loop perspective. Unlike them, this paper proposes a novel meta learning mechanism to self-adaptively obtain the learning rates of the learner, which allocates different learning rates for different tasks and neural network layers.

2.2 Curriculum Learning

Compared with the general paradigm of machine learning without distinction, curriculum learning is proposed to imitate the process of human learning (Bengio et al., 2009). It advocates that the model should start learning from easy instances and gradually advance to hard instances. Curriculum learning has been widely applied in many fields, e.g., CV (Guo et al., 2018; Jiang et al., 2014) and NLP (Platanios et al., 2019; Tay et al., 2019). Furthermore, curriculum learning can also be applied in other technical frameworks, e.g., reinforcement learning (Florensa et al., 2017; Narvekar et al., 2017; Ren et al., 2018), graph learning (Gong et al., 2019; Qu et al., 2018) and continual learning (Wu et al., 2021). In this paper, we extend the traditional instance-oriented curriculum learning to a task-oriented one, which gradually learns from different tasks with increasing difficulties.

3 Notations

In meta learning based few-shot text classification, two datasets are given: D_{train} and D_{test}, which have disjoint label sets. T tasks are sampled from D_{train} and the t-th task ($t \in [1, T]$), $Task_t$, consists of a support set S_t and a query set Q_t. Following the setting (Gao et al., 2019), we adopt C-way K-shot (hereinafter denoted as CwKs) for few-shot text classification, meaning S_t contains C classes and each class has K labeled instances. Thus, S_t can be formulated as $S_t = \{(x_t^i, y_t^i)\}_{i=1}^{C\times K}$, where x_t^i denotes the i-th piece of text in $Task_t$ and y_t^i is its class label. Furthermore, x_t^i contains M_t^i words (hereinafter simplified as M if not causing any confusion) and the m-th word ($m \in [1, M]$) in x_t^i denotes as $w_{t,m}^i$. Thus, $x_t^i = \{w_{t,m}^i\}_{m=1}^{M}$. x_t^i additionally includes a head entity h_t^i and a tail

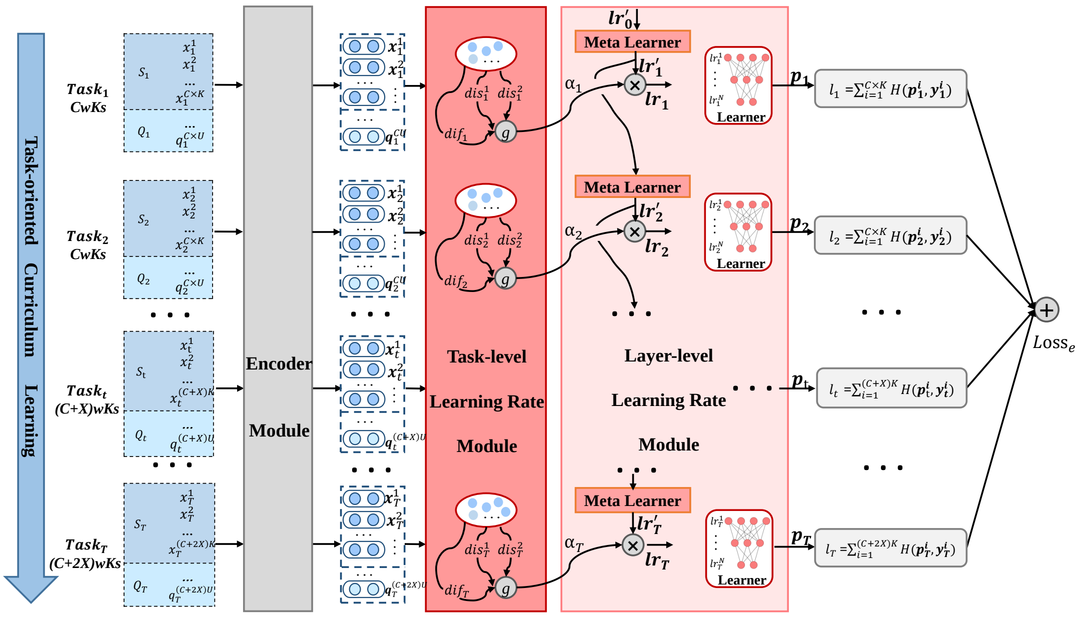

Figure 1: The diagram of the MetaSLRCL framework.

entity o_t^i in relation classification. Moreover, the query set Q_t contains U_t unlabeled instances for each class in S_t, where the i-th instance denotes q_t^i. Q_t can thus be formulated as $Q_t = \{q_t^i\}_{i=1}^{C\times U}$.

4 The MetaSLRCL Framework

MetaSLRCL is a generic framework, where few-shot learning models of different categories (i.e., model fine-tuning based, metric learning based, and meta learning based) can be adopted as the learner. As shown in Figure 1, MetaSLRCL consists of three modules coupled with a task-oriented curriculum learning mechanism.

The Encoder Module. This module maps the instances into the semantic space as embeddings via the encoder network.

The Task-level Learning Rate Module. This module calculates the task-level learning rate via the number of training classes and the distance between different instances in the support set.

The Layer-level Learning Rate Module. In this module, the layer-level learning rate is self-adaptively obtained based on the meta learning mechanism. This module contains two main parts: the learner as the classifier and the meta learner above the learner, which allocates learning rates for different network layers of the learner.

The Task-oriented Curriculum Learning Mechanism. It enables MetaSLRCL to gradually learn from tasks with more and more classes, thus with increasing difficulties, to make the meta learner achieve a better generalization ability.

4.1 The Encoder Module

The encoder module encodes each instance x_t^i into an embedding $\boldsymbol{x}_t^i$. This module consists of two parts, i.e., the embedding part and the encoding part.

In the embedding part, the semantic embeddings $\boldsymbol{w}_{t,m}^i$ for each word $w_{t,m}^i$ in x_t^i is obtained by looking up table. In this paper, we employ GloVe (Pennington et al., 2014) to obtain word embeddings for its fast training and remarkable performance even with small corpus. In the encoding part, the CNN encoder is employed because of its good performance and time efficiency to derive the instance embedding $\boldsymbol{x}_t^i$ of B dimension of x_t^i based on the word embeddings $\{\boldsymbol{w}_{t,m}^i\}_{m=1}^M$. CNN slides a conventional kernel with a window of size k, over the input embeddings to get the output hidden embeddings,

$$\boldsymbol{h}_{t,m}^i = \mathrm{Con}\left(\boldsymbol{w}_{t,m-\frac{k-1}{2}}^i, ..., \boldsymbol{w}_{t,m+\frac{k-1}{2}}^i\right), \quad (1)$$

where $\mathrm{Con}\left(\cdot\right)$ is a conventional operation.

A max pooling operation is then applied over these hidden embeddings to output the final instance embedding $\boldsymbol{x}_t^i$ as follows:

$$[\boldsymbol{x}_t^i]_b = max\left\{[\boldsymbol{h}_{t,1}^i]_b, ..., [\boldsymbol{h}_{t,M}]_b\right\}, \quad (2)$$

where $[\cdot]_b$ is the b-th value of a vector ($b \in [1, B]$).

4.2 The Task-level Learning Rate Module

This module is designed to self-adaptively get different learning rates for different tasks. In the context of few-shot learning, it is necessary for a model to converge within only a few steps (Finn et al., 2017). Intuitively, for easy tasks, large learning rates enable the model to converge fast. However, for hard tasks, relatively small learning rates are preferred so as to help the model carefully search for the optimal parameters in the complex search space. In this module, the difficulty of a task is defined as the learning difficulty, measured in terms of the number of training classes and the distance between different instances in the support set.

In more detail, the learning difficulty of a task is related to the number of classes in meta training. If the number, C, of training classes, of $Task_t$ is equal to that of its meta test classes, $C^{'}$, its difficulty coefficient dif_t is set to 1. If C is larger than $C^{'}$, indicating that it is a harder task, dif_t is increased. Otherwise, it is reduced. dif_t can be formally calculated as follows:

$$dif_t = 1 + \gamma \left(C - C^{'}\right), \tag{3}$$

where γ is a coefficient within $[0, 1]$.

The distance between different instances can be measured from two aspects, namely, the average intra-class distance dis_t^1 and the average inter-class distance dis_t^2. The closer the intra-class distance and the farther the inter-class distance, the easier the task. Both of them are measured by the Euclidean distance function $d(\cdot, \cdot)$. Specifically, dis_t^1 is calculated by

$$dis_t^1 = \frac{1}{D_t^1} \sum_{v=1}^{D_t^1} d\left(\boldsymbol{x}_t^i, \boldsymbol{x}_t^j\right), \tag{4}$$

where $\boldsymbol{x}_t^i$ and $\boldsymbol{x}_t^j$ $(i \neq j)$ belong to the same class; $D_t^1 = CK(K-1)/2$, denoting the number of pairs $(\boldsymbol{x}_t^i, \boldsymbol{x}_t^j)$. dis_t^2 is calculated as follows:

$$dis_t^2 = \frac{1}{D_t^2} \sum_{v=1}^{D_t^2} d\left(\boldsymbol{x}_t^i, \boldsymbol{x}_t^j\right), \tag{5}$$

where $\boldsymbol{x}_t^i$ and $\boldsymbol{x}_t^j$ belong to different classes and $D_t^2 = CK(C-1)K/2$. Therefore, the learning rate $\alpha_t^{'}$ of $Task_t$ can be calculated as

$$\alpha_t^{'} = \frac{dis_t^2}{dif_t \cdot dis_t^1}. \tag{6}$$

As aforesaid, larger learning rates are preferred for easier tasks. Therefore, Equation (6) means a larger $\alpha_t^{'}$ is obtained with dis_t^2 increasing, as well as dif_t and dis_t^1 decreasing, which indicates an easier task. Otherwise, a smaller $\alpha_t^{'}$ represents a harder task.

As the task-level learning rate is required to multiply the layer-level one in Equation (12), it should be larger than 1 for easier tasks and smaller than 1 for harder tasks. Therefore, we formulate the task-level learning rate $\alpha_t \in [\beta, 1+\beta]$ by function $g(\cdot)$ as

$$\alpha_t = g\left(\alpha_t^{'}\right) = nor\left(\alpha_t^{'}\right) + \beta, \tag{7}$$

where $nor(\cdot)$ is the min-max normalization function to normalize $\alpha_t^{'}$ between 0 and 1. In this paper, the bias β is set to 0.5.

4.3 The Layer-level Learning Rate Module

As aforementioned, this module contains a learner and a meta learner.

4.3.1 The Learner

In text classification, the learner is actually a classifier. Existing models of different categories can be employed as the learner, e.g., BERT (Kenton and Toutanova, 2019), PN (Snell et al., 2017) and ML-MAN (Ye and Ling, 2019), which are pre-trained. By inputting the embedding $\boldsymbol{x}_t^i$, the learner with the learning rate $\boldsymbol{lr}_t$, which is obtained by Equation (12), outputs the predicted probability distribution, $\boldsymbol{p}_t^i$, to different classes. Formally, $\boldsymbol{p}_t^i$ is calculated as follows:

$$\boldsymbol{p}_t^i = Learner\left(\boldsymbol{x}_t^i, \boldsymbol{lr}_t\right). \tag{8}$$

The loss of the learner is defined as l_t, which is calculated by the cross entropy function $H(\cdot, \cdot)$ as

$$l_t = \sum_{i=1}^{C \times K} H\left(\boldsymbol{p}_t^i, \boldsymbol{y}_t^i\right), \tag{9}$$

where $\boldsymbol{y}_t^i$ is the ground truth distribution of $\boldsymbol{x}_t^i$ to different classes.

4.3.2 The Meta Learner

The meta learner allocates different learning rates for different network layers. Let θ be its parameters. Given the layer-level learning rate $\boldsymbol{lr}_{t-1}^{'}$ of N dimension corresponding to $Task_{t-1}$ of the learner, the hidden state $\boldsymbol{hs}_t$ of the meta learner to $Task_t$ is calculated upon $\boldsymbol{lr}_{t-1}^{'}$ and its last hidden state $\boldsymbol{hs}_{t-1}$ as

$$\boldsymbol{hs}_t = MetaLearner_\theta\left(\boldsymbol{hs}_{t-1}, \boldsymbol{lr}_{t-1}^{'}\right). \tag{10}$$

Then, the layer-level learning rate $\boldsymbol{lr}_t^{'}$ of $Task_t$ is obtained upon the state $\boldsymbol{hs}_t$ as

Algorithm 1 The Training Pro. of Meta Learning.

Given a set of labeled training data D_{train}
Init parameters of the meta leaner as θ
Given the initial learning rate $\boldsymbol{lr}_0^{'}$
For $e \rightarrow 1$ to E do:
  Given a pre-trained learner with $\boldsymbol{lr}_0^{'}$
  For $t \rightarrow 1$ to T do:
    Given a task $Task_t$ sampled from D_{train}
    $\boldsymbol{hs}_t \leftarrow MetaLearner_\theta \left(\boldsymbol{hs}_{t-1}, \boldsymbol{lr}_{t-1}^{'}\right)$
    $\boldsymbol{lr}_t^{'} \leftarrow \sigma\left(\boldsymbol{W}\boldsymbol{hs}_t + \boldsymbol{b}\right)$
    $\boldsymbol{lr}_t \leftarrow \alpha_t \boldsymbol{lr}_t^{'}$
    Train the learner with $\boldsymbol{lr}_t$ on $Task_t$ in one step
    Compute the loss l_t
    If $t = T$, calculate the loss $Loss_e$ by summing up l_t
  Update θ using $Loss_{e-1}$- $Loss_e$

$$\boldsymbol{lr}_t^{'} = \sigma\left(\boldsymbol{W}\boldsymbol{hs}_t + \boldsymbol{b}\right), \tag{11}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are parameters of a fully-connected layer and $\sigma(\cdot)$ is the Sigmoid activation function.

By multiplying the task-level learning rate α_t, the final learning rate is obtained as

$$\boldsymbol{lr}_t = \alpha_t \boldsymbol{lr}_t^{'}. \tag{12}$$

The loss of the meta learner in the e-th iteration ($e \in [1, E]$), $Loss_e$ is calculated by summing up the losses l_t of all tasks from the learner as

$$Loss_e = \sum_{t=1}^{T} l_t. \tag{13}$$

Finally, θ is updated by minimizing the difference between the loss in the last iteration and the current loss, which makes the meta learner converge faster, through applying gradient-based optimization. The training process of meta learning is shown in Algorithm 1.

4.4 The Task-oriented Curriculum Learning Mechanism

To get better generalization performance to new tasks, MetaSLRCL introduces a task-oriented curriculum learning mechanism to the meta training period. The original curriculum learning mechanism learns from instances with gradually increasing difficulties in a step-by-step manner. However, in the context of meta learning, we need to pay more attention to tasks with different difficulties. It is acknowledged that when the number of classes in a task increases, its difficulty increases accordingly. For example, a 10w1s task is harder than a 5w1s one. Therefore, a three-stage process with increasing difficulties is carried out with the number of classes increasing from C to C+X and further to C+2X (hereinafter denoting the process as C-(C+X)-(C+2X)), making the meta learner train tasks from easy to hard. Besides, a previous study (Munkhdalai and Yu, 2017) found that the models trained on harder tasks, but tested with relatively easier tasks may achieve better performance, as compared with those models which are trained and tested on tasks with the same difficulty configuration. Thus, in this paper we set that the average difficulty of tasks in the meta training period is always higher than that in the meta test period to get better performance in test tasks.

5 Experiments

5.1 Datasets and Evaluation Metrics

Parameters	Value
γ	0.1
β	0.5
k	3
word emb. dim.	50
max sentence length	40
hidden layer dim.	230
LSTM hidden size	100
initial learning rate	[$7e^{-3}$, $6e^{-3}$, $5e^{-3}$, $4e^{-3}$]
batch size	1
T	600
E	50
dropout	0.2

Table 1: The parameter setting in MetaSLRCL.

To verify the effectiveness of the MetaSLRCL framework, we conduct experiments on three different types of text classification, i.e., relation classification, news classification, and topic classification with three representative benchmark datasets. For relation classification, we choose a typical few-shot learning dataset, FewRel (Han et al., 2018). Note that the FewRel dataset used in this paper has only 80 classes, thus marked as FewRel80, because 20 classes of the original FewRel dataset for test are not publicly available. We randomly divide FewRel80 into three subsets containing 50, 10 and 20 classes for training, validation and test, respectively. For news classification, we choose the representative dataset, 20Newsgroup (Dadgar et al., 2016) with 20 news classes. As 20Newsgroup lacks standard splits in few-shot learning, we randomly divide it into subsets with 14 and 6 classes for training and test, respectively. For topic classification, the DBPedia Ontology (Zhang et al., 2015) dataset is a classic one with 14 topic classes. Similarly, we randomly partition it into 8 classes

Dataset: FewRel80					
Model		5w1s	5w5s	10w1s	10w5s
model fine-tuning based	BERT	0.5762	0.7109	0.5233	0.5480
	MetaSLRCL+BERT	**0.6347**	**0.7601**	**0.5672**	**0.5988**
metric learning based	PN_HATT	0.7319	0.8703	0.6114	0.7632
	MetaSLRCL+PN_HATT	**0.7675**	**0.8929**	**0.6507**	**0.8067**
meta learning based	MLMAN	0.7957	0.9119	0.6903	0.8516
	MetaSLRCL+MLMAN	**0.8182**	**0.9150**	**0.7084**	**0.8519**
Dataset: 20Newsgroup					
Model		3w1s	3w5s	6w1s	6w5s
model fine-tuning based	BERT	0.7417	0.8198	0.5876	0.7107
	MetaSLRCL+BERT	**0.7689**	**0.8476**	**0.6187**	**0.7426**
metric learning based	PN	0.8463	0.9614	0.7052	0.8887
	MetaSLRCL+PN	**0.8680**	**0.9843**	**0.7217**	**0.9264**
meta learning based	MAML	0.7612	0.8405	0.6143	0.7451
	MetaSLRCL+MAML	**0.7824**	**0.8599**	**0.6465**	**0.7738**
Dataset: DBPedia Ontology					
Model		3w1s	3w5s	6w1s	6w5s
model fine-tuning based	BERT	0.7609	0.8256	0.6118	0.7589
	MetaSLRCL+BERT	**0.7928**	**0.8598**	**0.6540**	**0.7990**
metric learning based	PN	0.8428	0.9520	0.7070	0.8896
	MetaSLRCL+PN	**0.8683**	**0.9799**	**0.7301**	**0.9104**
meta learning based	MAML	0.7778	0.8571	0.6434	0.8093
	MetaSLRCL+MAML	**0.8110**	**0.8911**	**0.6786**	**0.8359**

Table 2: The overall results on three benchmark datasets: FewRel80, 20Newsgroup and BDPedia Ontology.

and 6 classes for training and test, respectively.

We set up four configurations, namely, 5w1s, 5w5s, 10w1s and 1w5s, on FewRel80. Four settings are considered for the 20Newsgroup and DBPedia Ontology datasets, i.e., 3w1s, 3w5s, 6w1s and 6w5s. Following the previous study in (Obamuyide and Vlachos, 2019), average accuracy upon 5 runs is adopted as the evaluation metric.

5.2 Implementation Details and Parameters Setting

Table 1 presents the parameter setting of MetaSLRCL. For the encoder module, CNN is employed as the encoder and the word embeddings pre-trained in GloVe (Pennington et al., 2014) are adopted as the initial embeddings. More specifically, we choose the embedding set of GloVe trained on Wikipedia 2014 + Gigaword 5, which contains 6B tokens and 400K words. The word embeddings are of 50 dimensions. For the parameters of CNN, we follow the settings used in (Zeng et al., 2014). For the layer-level learning rate module, LSTM is selected as the meta learner, because of its simple implementation, fast training speed and remarkable performance. Furthermore, for the curriculum learning, we choose one setting with best performance on each dataset, specifically, 10-15-20 on FewRel80, 7-9-11 on 20Newsgroup and 5-6-7 on DBPedia Ontology.

5.3 Baseline Models

As MetaSLRCL is a generic framework, it can employ different few-shot learning models as its learner. Therefore, in the experiments, we adopt the representative and state-of-the-art (SOTA) models of the aforesaid different categories as the learner of MetaSLRCL in order to verify its effectiveness on different tasks. These models are also adopted as the baselines for performance comparison. It should be particularly mentioned that for the sake of space limitation, for each type of text classification and each category of the few-shot learning models, the experimental results of only the baseline models (e.g., MAML) with the best performance and their MetaSLRCL counterparts (e.g., MetaSLRCL+MAML) will be presented. More specifically, for relation classification, the baseline models include: 1) BERT-base-uncased (Kenton and Toutanova, 2019), a widely adopted model of model fine-tuning based category; 2) PN_HATT (Gao et al., 2019), the SOTA metric learning based model especially for relation classification; 3) MLMAN (Ye and Ling, 2019), the SOTA model in few-shot relation classification. For news classification and topic classifica-

Model		5w1s	5w5s	10w1s	10w5s
model fine-tuning based	MetaSLRCL+BERT	**0.6347**	**0.7601**	**0.5672**	**0.5988**
	SLR+BERT	0.6174	0.7456	0.5532	0.5851
	CL+BERT	0.5904	0.7263	0.5370	0.5615
metric learning based	MetaSLRCL+PN_HATT	**0.7675**	**0.8929**	**0.6507**	**0.8067**
	SLR+PN_HATT	0.7592	0.8831	0.6435	0.7982
	CL+PN_HATT	0.7380	0.8719	0.6152	0.7792
meta learning based	MetaSLRCL+MLMAN	**0.8182**	**0.9150**	**0.7084**	0.8519
	SLR+MLMAN	0.8103	0.9145	0.7059	**0.8541**
	CL+MLMAN	0.8167	0.9136	0.7042	0.8507

Table 3: The results of the ablation study on SLR and CL on FewRel80.

Model		5w1s	5w5s	10w1s	10w5s
model fine-tuning based	SLR+BERT	**0.6174**	**0.7456**	**0.5532**	**0.5851**
	SLRL+BERT	0.6145	0.7412	0.5509	0.5823
	SLRT+BERT	0.5771	0.7148	0.5261	0.5502
metric learning based	SLR+PN_HATT	**0.7592**	**0.8831**	**0.6435**	**0.7982**
	SLRL+PN_HATT	0.7578	0.8811	0.6414	0.7956
	SLRT+PN_HATT	0.7354	0.8723	0.6137	0.7648
meta learning based	SLR+MLMAN	**0.8103**	**0.9145**	**0.7059**	**0.8541**
	SLRL+MLMAN	0.8095	0.9139	0.7051	0.8537
	SLRT+MLMAN	0.7982	0.9125	0.6931	0.8522

Table 4: The results of the ablation study on SLRs on FewRel80.

tion, the baseline models are the same, including: 1) BERT-base-uncased, for the same reason; 2) PN (Snell et al., 2017), a widely adopted metric learning based model; 3) MAML (Finn et al., 2017), a widely adopted meta learning based model.

5.4 Main Results

Table 2 presents the main results, where we can see that all of the MetaSLRCL models with BERT, PN_HATT, MLMAN, PN and MAML as their learners consistently outperform those corresponding baselines on all datasets. The accuracy of the model fine-tuning based and metric learning based MetaSLRCL models increases by 4-6% and 2-4% on FewRel80, respectively. However, for MetaSLRCL+MLMAN, its performance is improved less than those of the former two categories; But it still achieves the best results. Moreover, all kinds of MetaSLRCL models are observed accuracy promotion by 2-4% compared to the baselines on the majority of few-shot tasks on 20Newsgroup and DBPedia Ontology. In short, these experimental results convincingly suggest that MetaSLRCL is effective for different tasks on different datasets and with different models.

5.5 Ablation Studies

5.5.1 SLR and CL in MetaSLRCL

In this subsection, we conduct ablation studies to investigate the effectiveness of both Self-adaptive Learning Rate (SLR) and Curriculum Learning (CL), as well as their impacts on the performance of MetaSLRCL. For the sake of space limitation, only the results on FewRel80 are presented. As shown in Table 3, the performance of all ablated models without SLR and CL consistently falls, except MLMAN on the 10w5s task. For each type of the models in this table, we adopt the same CL setting on different tasks, with which the MetaSLRCL enhanced model exhibits best performance on most of them. Therefore, for the MLMAN models, the 10-15-20 CL setting is selected, because under this CL setting the MetaSLRCL+MLMAN model achieves the best results on the 5w1s, 5w5s and 10w1s tasks. Nevertheless, on the 10w5s task, MetaSLRCL+MLMAN obtains its best performance with the CL setting of 15-20-25. For this reason, SLR+MLMAN exceptionally outperforms MetaSLRCL+MLMAN on the 10w5s task. The general results in Table 3 indicate that both SLR and CL contribute to the effectiveness of MetaSLRCL. Besides, it can be observed that SLR is more important to MetaSLRCL than CL, because of the larger performance improvement. Similar phenomena can be observed on the other datasets, 20Newsgroup and DBPedia Ontology.

5.5.2 SLRs for Tasks and Network Layers

In MetaSLRCL, SLR consists of two subsets, the Self-adaptive Learning Rates for different

Model	5w1s	5w5s
Adadelta+BERT	0.5825	0.7232
RMSProp+BERT	0.5887	0.7203
Adam+BERT	0.5943	0.7261
SLR+BERT	**0.6174**	**0.7456**
Adadelta+PN_HATT	0.7386	0.8612
RMSProp+PN_HATT	0.7327	0.8446
Adam+PN_HATT	0.7101	0.8300
SLR+PN_HATT	**0.7592**	**0.8831**
Adadelta+MLMAN	0.7995	0.9063
RMSProp+MLMAN	0.8007	0.9087
Adam+MLMAN	0.8027	0.9108
SLR+MLMAN	**0.8103**	**0.9145**

Table 5: The results of different models with SLR and other self-adaptive learning rate mechanisms on FewRel80.

Tasks (SLRT) and different neural network Layers (SLRL). As shown in Table 4, the performance of all models without SLRT and SLRL consistently decreases, indicating that both SLRT and SLRL are important to the effectiveness of SLR. However, the models with SLRL outperform those with SLRT. That means, although both task-level and layer-level learning rates work, the layer-level ones are more important and effective to the performance of models than their counterparts.

5.6 SLR Comparing with Other Self-Adaptive Learning Rate Methods

We also conduct some experiments to compare our SLR with other self-adaptive learning rate mechanisms, i.e., Adadelta (Zeiler, 2012), RMSProp (Hinton et al., 2012) and Adam (Kingma and Ba, 2014), on FewRel80. The parameters of these methods are tuned on our dataset. The experimental results are shown in Table 5. It can be noted that, the models with our SLR outperform all the others, which indicates that our SLR is more effective than the others. Moreover, as compared with Adadelta, the performance of RMSProp and Adam are unstable when coupled with different models, i.e., BERT, PN_HATT, and MLMAN. Differently, our SLR exhibits consistently the best performance in all cases, indicating that our SLR is more robust than the others when applied to different models.

As mentioned in Section 2, there have already been some models in CV, which explore self-adaptive learning rates, e.g., MAML++ and ALFA. We experimentally compare our SLR in the MetaSLRCL framework with them at the method level. The experimental results are shown in Table 6. Note that for fair comparison, the same initial learning rate as ours is adopted. As we can see, the accuracy of MAML++ and ALFA is lower than all of the MetaSLRCL models with our SLR. It suggests that although MAML++ and ALFA achieve superior performance in CV, our SLR outperforms them on text classification.

	Model	5w1s	5w5s
CV	MAML++	0.5823	0.6954
	ALFA	0.6009	0.7137
Ours	SLR+BERT	0.6174	0.7456
	SLR+PN_HATT	0.7592	0.8831
	SLR+MLMAN	**0.8103**	**0.9145**

Table 6: The results of self-adaptive learning rate models in CV and our SLR on FewRel80.

Model	5w1s	5w5s
SLR+5-10-15+BERT	0.6285	0.7498
SLR+10-15-20+BERT	**0.6347**	**0.7601**
SLR+15-20-25+BERT	0.6315	0.7581
SLR+20-25-30+BERT	0.6239	0.7475
SLR+5-10-15+PN_HATT	0.7562	0.8836
SLR+10-15-20+PN_HATT	0.7565	**0.8929**
SLR+15-20-25+PN_HATT	**0.7675**	0.8877
SLR+20-25-30+PN_HATT	0.7645	0.8926
SLR+5-10-15+MLMAN	0.8102	0.9135
SLR+10-15-20+MLMAN	**0.8182**	0.9150
SLR+15-20-25+MLMAN	0.8133	**0.9161**
SLR+20-25-30+MLMAN	0.8046	0.9146

Table 7: The results of different CL settings on FewRel80.

5.7 Different CL Settings

We also conduct experiments to evaluate the impact of the CL mechanism. Specifically, we set up four training settings for each task on FewRel80, namely, 5-10-15, 10-15-20, 15-20-25 and 20-25-30. For the sake of space limitation, only results on 5w1s and 5w5s are shown in Table 7, which demonstrate that all the best results are obtained at two settings, 10-15-20 and 15-20-25. This may be due to the following reason: the 5-10-15 configuration is the simplest one, which does not reach the difficulty to get the best performance of a model, whilst the 20-25-30 configuration is too hard and the learner cannot be well trained at the training period and thus cannot work well at the test period.

Furthermore, four training settings, namely, 3-5-7, 5-7-9, 7-9-11 and 9-11-13 are examined on 20Newsgroup. Four training settings, i.e., 3-4-5, 4-5-6, 5-6-7 and 6-7-8 are also studied on DBPedia Ontology. Similar phenomena can be observed on these datasets. The results are not presented due to space limitation.

6 Conclusion and Future Work

In this paper, we proposed a novel meta learning framework, called MetaSLRCL, for few-shot text classification. MetaSLRCL can self-adaptively obtain different learning rates for different tasks and different network layers. Moreover, a task-oriented curriculum learning mechanism is introduced into few-shot learning to achieve a better generalization ability for the meta learner. MetaSLRCL is evaluated with three typical types of text classification, relation classification, news classification and topic classification, on three benchmark datasets, namely, FewRel80, 20Newsgroup and DBPedia Ontology, respectively. Experimental results demonstrate superior performance of MetaSLRCL on all datasets. In the future, we will explore few-shot learning under the unbalance learning scenarios because they are ubiquitous in the real world.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under grants U1911401, 62002341, and 61772501, the GFKJ Innovation Program, and the Lenovo-CAS Joint Lab Youth Scientist Project.

References

Antreas Antoniou, Harri Edwards, and Amos Storkey. 2019. How to train your maml. In *Seventh International Conference on Learning Representations*. 2.1

Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. 2020. Meta-learning with adaptive hyperparameters. *Advances in Neural Information Processing Systems*, 33:20755–20765. 2.1

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. 1, 2.2

Seyyed Mohammad Hossein Dadgar, Mohammad Shirzad Araghi, and Morteza Mastery Farahani. 2016. A novel text mining approach based on tf-idf and support vector machine for news classification. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 112–116. IEEE. 5.1

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR. 1, 2.1, 4.2, 5.3

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. 2017. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR. 2.2

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6407–6414. 1, 3, 5.3

Chen Gong, Jian Yang, and Dacheng Tao. 2019. Multimodal curriculum learning over graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(4):1–25. 2.2

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150. 2.2

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809. 5.1

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural networks for machine learning, Coursera lecture 6e*, page 13. 5.6

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. 1, 2.1

Muhammad Abdullah Jamal and Guo-Jun Qi. 2019. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11719–11727. 2.1

Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 547–556. 2.2

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186. 4.3.1, 5.3

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv e-prints*, pages arXiv–1412. 5.6

Fei-Fei Li et al. 2003. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1134–1141. IEEE. 1

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011. 1

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR. 1, 2.1, 4.4

Akihiro Nakamura and Tatsuya Harada. 2019. Revisiting fine-tuning for few-shot learning. *arXiv preprint arXiv:1910.00216*. 1, 2.1

Sanmit Narvekar, Jivko Sinapov, and Peter Stone. 2017. Autonomous task sequencing for customized curriculum design in reinforcement learning. In *IJCAI*, pages 2536–2542. 2.2

Abiola Obamuyide and Andreas Vlachos. 2019. Model-agnostic meta-learning for relation classification with limited supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5873–5879. 5.1

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 (EMNLP)*, pages 1532–1543. 4.1, 5.2

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of NAACL-HLT*, pages 1162–1172. 2.2

Meng Qu, Jian Tang, and Jiawei Han. 2018. Curriculum learning for heterogeneous star network embedding via deep reinforcement learning. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 468–476. 2.2

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. 2019. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*. 2.1

Zhipeng Ren, Daoyi Dong, Huaxiong Li, and Chunlin Chen. 2018. Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning. *IEEE transactions on neural networks and learning systems*, 29(6):2216–2226. 2.2

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*. 2.1

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4080–4090. 1, 2.1, 4.3.1, 5.3

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931. 2.2

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638. 1, 2.1

Jiawei Wu, Wenhan Xiong, and William Yang Wang. 2019. Learning to learn and predict: A meta-learning approach for multi-label classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4354–4364. 2.1

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Gholamreza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10363–10369. 2.2

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-level matching and aggregation network for few-shot relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2872–2881. 4.3.1, 5.3

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*. 5.6

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. 1, 5.2

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657. 5.1

Jianming Zheng, Fei Cai, Wanyu Chen, Wengqiang Lei, and Honghui Chen. 2021. Taxonomy-aware learning for few-shot event detection. In *Proceedings of the Web Conference 2021*, pages 3546–3557. 1