Knowledge Guided Metric Learning for Few-Shot Text Classification

Dianbo Sui[1,2], **Yubo Chen**[1,2], **Binjie Mao**[1,2], **Delai Qiu**[3], **Kang Liu**[1,2], **Jun Zhao**[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Beijing Unisound Information Technology Co., Ltd.
{dianbo.sui, yubo.chen, binjie.mao, kliu, jzhao}@nlpr.ia.ac.cn
qiudelai@unisound.com

Abstract

Humans can distinguish new categories very efficiently with few examples, largely due to the fact that human beings can leverage knowledge obtained from relevant tasks. However, deep learning based text classification model tends to struggle to achieve satisfactory performance when labeled data are scarce. Inspired by human intelligence, we propose to introduce external knowledge into few-shot learning to imitate human knowledge. A novel parameter generator network is investigated to this end, which is able to use the external knowledge to generate different metrics for different tasks. Armed with this network, similar tasks can use similar metrics while different tasks use different metrics. Through experiments, we demonstrate that our method outperforms the SoTA few-shot text classification models.

1 Introduction

Humans are adept at quickly learning from a small number of examples. This motivates research of few-shot learning (Vinyals et al., 2016; Finn et al., 2017), which aims to recognize novel categories from very few labeled examples.

The key challenge in few-shot learning is to make full use of the limited labeled examples to find the "right" generalizations. Metric-based approaches (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018; Li et al., 2019; Zhang et al., 2020) are effective ways to address this challenge, which learn to represent examples in an appropriate feature space and use a distance metric to predict labels. However, directly employing metric-based approaches in text classification faces a problem that tasks are diverse and significantly different from each other, since words that are highly informative for one task may not be relevant for other tasks (Bao et al., 2019). Therefore, a single metric is insufficient to cope with all these tasks in few-shot text classification (Yu et al., 2018).

To adapt metric learning to significantly diverse tasks, we propose a knowledge guided metric learning method. This method is inspired by the fact that human beings approach diverse tasks armed with knowledge obtained from relevant tasks (Lake et al., 2017). We use external knowledge from the knowledge base (KB) to imitate human knowledge, whereas the role of external knowledge has been ignored in previous methods (Yu et al., 2018; Bao et al., 2019; Geng et al., 2019, 2020). In detail, we resort to distributed representations of the KB instead of symbolic facts, since symbolic facts face the issues of poor generalization and data sparsity. Based on such KB embeddings, we investigate a novel parameter generator network (Ha et al., 2016; Jia et al., 2016) to generate task-relevant relation network parameters. With these generated parameters, the task-relevant relation network is able to apply diverse metrics to diverse tasks and ensure that similar tasks use similar metrics while different tasks use different metrics.

In summary, the major contributions of this paper are:

- Inspire by human intelligence, we present the first approach that introduces external knowledge into few-shot learning.
- A novel parameter generator network based on external knowledge is proposed to generate diverse metrics for diverse tasks.
- Experimental results on the public dataset show that our model significantly outperforms previous methods.

2 Problem Setting

Few-shot classification aims at training a model that can recognize novel classes from very few labeled examples. The training and testing of the model are conducted on two datasets (training set and test set) with no overlapped classes. At

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3266–3271
June 6–11, 2021.

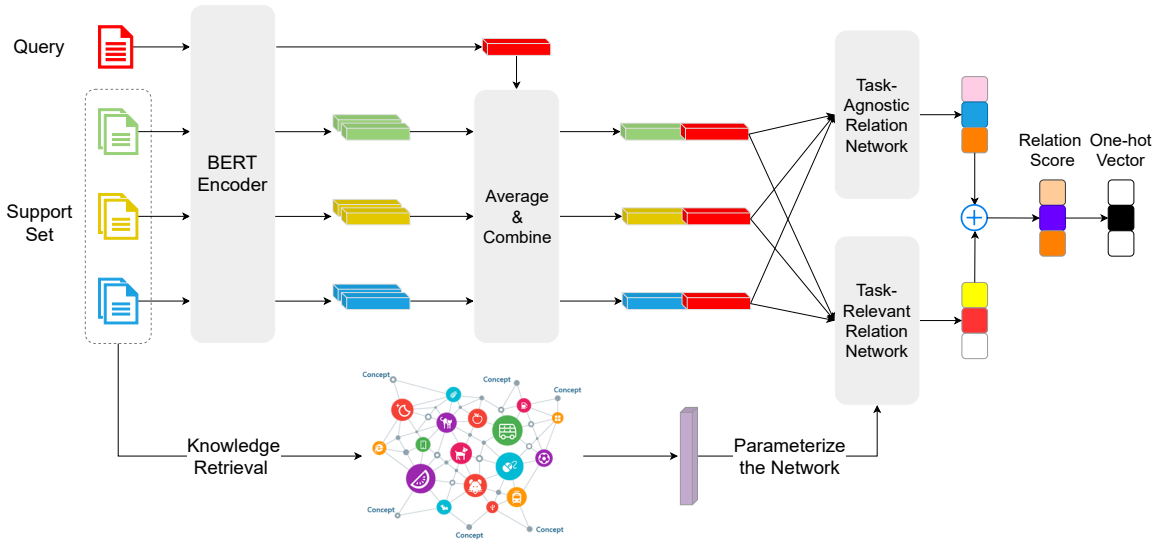

Figure 1: The main architecture for a C-way N-shot (C = 3, N = 2) problem with one query example.

both the training and test stages, the labeled examples are called the support set, which serves as a meta-training set and the meta-testing examples are called the query set. If the support set contains N labeled examples for each of C unique classes, the few-shot problem is called C-way N-shot. To guarantee a good generalization performance at test time, the training and evaluation of the model are accomplished by episodically sampling the support set and the query set (Vinyals et al., 2016). More concretely, in each meta-training iteration, an episode is formed by randomly selecting C classes from the training set with N labeled examples for each of the C classes to serve as the support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{C\times N}$, as well as a fraction of the remainder of those C classes' examples to act as the query set $\mathcal{Q} = \{(x_i, y_i)\}_{i=C\times N+1}^{C\times N+m}$, where x_i and $y_i \in \{1, ..., C\}$ are the sentence and its label, and m is the number of query samples. The model is trained on the support set $\mathcal{S}$ to minimize the loss of its predictions over the query set $\mathcal{Q}$. This training procedure is iteratively carried out episode by episode until convergence.

3 Methodology

3.1 Sentence Embedding Network

In this network, a pre-trained BERT (Devlin et al., 2019) encoder is used to model sentences. Given an input text $x_i = ([\text{CLS}], w_1, w_2, ..., w_T, [\text{SEP}])$ as input, the output of BERT encoder is denoted as $\mathbf{H}(x_i) \in \mathbb{R}^{(T+2)\times d_1}$, where d_1 is the output dimension of the BERT encoder. We use the first token of the sequence (classification token) as the sentence representation, which is denote as $\mathbf{h}(x_i)$.

In meta-learning, the representation of each class is the mean vector of the embedded sentences belonging to its class,

$$\mathbf{c}_z = \frac{1}{|\mathcal{S}_z|} \sum_{(x_i, y_i)\in\mathcal{S}_z} \mathbf{h}(x_i) \in \mathbb{R}^{d_1} \quad (1)$$

where $\mathcal{S}_z$ denotes the set of sentences labeled with class z. Following Sung et al. (2018), we use concatenation operator to combine the query representation $\mathbf{h}(x_j)$ with the class representation $\mathbf{c}_z$.

$$\mathbf{p}_{z,j} = \text{concatenation}(\mathbf{c}_z, \mathbf{h}(x_j)) \in \mathbb{R}^{2d_1} \quad (2)$$

3.2 Knowledge Guided Relation Network

This module takes combined representation (shown in Equation 2) and the knowledge of the support set as input, and produces a scalar in range of 0 to 1 representing the similarity between the query sentence and the class representation, which is called relation score. Compared with the original relation network (Sung et al., 2018), we decompose the relation network into two parts, task-agnostic relation network and task-relevant relation network, in order to serve two purposes. Task agnostic relation network models a basic metric function, while task-relevant relation network adapts to diverse tasks.

Task-Agnostic Relation Network The task-agnostic relation network uses a learned unified metric for all tasks, which is the same with the original relation network (Sung et al., 2018). With this unified metric, C task-agnostic relation scores $r_{z,j}^{agn}$

are generated for modeling the relation between one query input x_j and the class representation c_z,

$$r_{z,j}^{agn} = RN^{agn}(\mathbf{p}_{z,j}|\boldsymbol{\theta}^{agn}) \in \mathbb{R}, \quad z = 1, 2, ..., C \tag{3}$$

where RN^{agn} denotes task-agnostic relation network and $\boldsymbol{\theta}^{agn}$ are learnable parameters.

Task-Relevant Relation Network The task-relevant relation network is able to apply diverse metrics for diverse tasks armed with external knowledge. In detail, for each support set $\mathcal{S}$ ($\mathcal{S}$ contains $C \times N$ labeled sentences), we retrieve a set of potentially relevant KB concepts $K(\mathcal{S})$, where each concept k_i is associated with KB embedding $\mathbf{e}_i \in \mathbb{R}^{d_2}$. (we will describe these processes in the following section). We average over these KB embeddings element by element to form the knowledge representation of this support set.

$$\mathbf{k}_{\mathcal{S}} = \frac{1}{|K(\mathcal{S})|} \sum_{k_i \in K(\mathcal{S})} \mathbf{e}_i \in \mathbb{R}^{d_2} \tag{4}$$

Then we use this knowledge representation to generate task-relevant relation network parameters,

$$\boldsymbol{\theta}^{rel} = \mathbf{M} \cdot \mathbf{k}_{\mathcal{S}} \in \mathbb{R}^{d_3} \tag{5}$$

where $\mathbf{M} \in \mathbb{R}^{d_3 \times d_2}$ are learnable parameters and d_3 denotes the number of parameters of the task-relevant relation network.

With these generated parameters, we use the task-relevant network to generate C task-relevant relation scores $r_{z,j}^{rel}$ for the relation between one query input x_j and the class representation c_z,

$$r_{z,j}^{rel} = RN^{rel}(\mathbf{p}_{z,j}|\boldsymbol{\theta}^{rel}) \in \mathbb{R}, \quad z = 1, 2, ..., C \tag{6}$$

where RN^{rel} denotes task-relevant relation network. Finally, relation score is defined as:

$$r_{z,j} = Sigmoid(r_{z,j}^{agn} + r_{z,j}^{rel}) \tag{7}$$

where a sigmoid function is used to keep the score in a reasonable range. Following Sung et al. (2018), the network architecture of relation networks is two full-connected layers and mean square error (MSE) loss is used to train the model. The relation score is regressed to the ground truth: the matched pairs have similarity 1 and the mismatched pairs have similarity 0.

$$L = \sum_{z=1}^{C} \sum_{j=1}^{|\mathcal{Q}|} (r_{z,j} - \mathbf{1}(y_j == z)) \tag{8}$$

3.3 Knowledge Embedding and Retrieval

We use NELL (Carlson et al., 2010) as the KB, stored as (subject, relation, object) triples, where each triple is a fact indicating a specific relation between subject and object, e.g., (Intel, competes with, Nvidia).

Knowledge Embedding Since symbolic facts suffer from poor generalization and data sparsity, we resort to distributed a representation of triples. In detail, given any triple (s, r, o), vector embeddings of the subject s, the relation r and the object o are learned jointly such that the validity of the triple can be measured in the real number space. We adopt the BILINEAR model (Yang et al., 2015) to measure the validity of triples:

$$f(s, r, o) = \mathbf{s}^T diag(\mathbf{r})\mathbf{o} \in \mathbb{R} \tag{9}$$

where $\mathbf{s}, \mathbf{r}, \mathbf{o} \in \mathbb{R}^{d_2}$ are the embeddings associated with s, r, o, respectively, and $diag(\mathbf{r})$ is a diagonal matrix with the main diagonal given by the relation embedding $\mathbf{r}$. To learn these vector embeddings, a margin-based ranking loss is designed, where triples in the KB are adopted to be positive and negative triples are constructed by corrupting either subjects or objects.

Knowledge Retrieval Inspired by the previous studies (Yang and Mitchell, 2017; Yang et al., 2019), exact string matching (Charras and Lecroq, 2004) is used to recognize entity mentions from a given passage and link recognized entity mentions to subjects in KB. Then, we collect the corresponding objects (concepts) as candidates. After this retrieval process, we obtain a set of potentially relevant KB concepts, where each KB concept is associated with a KB embedding.

4 Experiment

4.1 Dataset

Our model is evaluated on the widely used ARSC (Blitzer et al., 2007) dataset, which comprises reviews for 23 types of products on Amazon. For each product domain, there are three different binary classification tasks. These buckets form 69 tasks in total. Following Yu et al. (2018), we select 12 tasks from four domains (Books, DVDs, Electronics, and Kitchen) as testing set, with only 5 examples as support set for each class.

4.2 Implementation Details

In our experiments, we use hugginface's implementation[1] of BERT (base version) and initialize parameters of the BERT encoding layer with pre-trained models officially released by Google[2]. To represent knowledge in NELL (Carlson et al., 2010), BILINEAR model (Yang et al., 2015) is implemented with the open-source framework OpenKE (Han et al., 2018) to obtain the embedding of entities and relations. The size of embeddings of entities and relations is set to 100. To train our model, We use Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.00001. All experiments are run with an NVIDIA GeForce RTX 2080 Ti.

4.3 Experiment Results

Baseline. We compare our method to the following baselines: (1) **Match Network** is a metric-based attention method for few-shot learning; (2) **Prototypical Network** is a metric-based method that uses sample averages as class prototypes; (3) **MAML** is an optimization-based method through learning to learn with gradients; (4) **Relation Network** is a metric-based method that leverages two full-connected layers as the distance metric and sums up sample vectors in the support set as class vectors; (5) **Graph Network** is a graph-based model that implements a task-driven message passing algorithm on the sample-wise level; (6) **ROBUSTTC-FSL** is an approach that combines adaptive metric methods by clustering the tasks; (7) **Induction Network** is a metric-based method by using dynamic routing to learn class-wise representations.

Model	Mean Acc
Matching Network (Vinyals et al., 2016)	65.73
Prototypical Network (Snell et al., 2017)	68.15
MAML (Finn et al., 2017)	78.33
Graph Network (Garcia and Bruna, 2017)	82.61
Relation Network (Sung et al., 2018)	83.07
ROBUSTTC-FSL (Yu et al., 2018)	83.12
Induction Network (Geng et al., 2019)	85.63
Ours	**87.93**

Table 1: Comparison of mean accuracy (%) on ARSC dataset. The scores of baselines are taken from Geng et al. (2019).

Analysis. Experiment results on ARSC are presented in Table 1. We observe that our method achieves the best results amongst all meta-learning models. Both Induction Network and Relation Network use a single metric to measure the similarity. Compared with these methods, we attribute the improvements of our model to the fact that our model can adapt to diverse tasks with diverse metrics. Compared with ROBUSTTC-FSL, our model leverages knowledge to get implicit task clusters and is trained in an end-to-end manner, which can mitigate error propagation.

4.4 Effectiveness of Introducing Knowledge

To analyze the contributions and effects of external knowledge in our approach, we perform some ablation and replacement studies, which are shown in Table 2. **Ablation** means that we delete the task-relevant relation network and the model is reduced to the original BERT-based relation network. We observe that ablation degrades performance. To exclude the factor of reduction in the number of parameters, we conduct a **replacement** experiment, in which we replace the task-relevant relation network with a task-agnostic relation network. We find that increasing the number of parameters can slightly improve performance, but there is still a big gap between our model. Therefore, we conclude that the effectiveness of our model is credited with introducing external knowledge rather than increasing the number of model parameters.

Model	Mean Acc
Ours	87.93
Ablation	86.09 (↓ 1.84)
Replacement	86.40 (↓ 1.53)

Table 2: Ablation and replacement studies of our model on ARSC dataset.

4.5 Different Strategies of Introducing Knowledge

To analyze different strategies of introducing knowledge in few-shot learning, we remove the task-relevant relation network, and replace the BERT encoder in our method with **KT-NET** encoder (Yang et al., 2019) and **K-BERT** encdoer (Liu et al., 2019). In the KT-NET encoder, an attention mechanism is used to adaptively fuse selected knowledge with BERT. In the K-BERT encoder, a knowledge-rich sentence tree is the input of the model. These methods both introduce knowledge

[1]https://huggingface.co/transformers
[2]https://github.com/google-research/bert

at the representation level [3], while our method injects knowledge at the task level. The result is shown in Table 3. Combined Table 2 and Table 3, we find that (1) introducing knowledge can improve the performance of few-shot text classification; (2) it is more effective to introduce knowledge at the task level rather than at the representation level.

Model	Mean Acc
Ours	87.93
K-BERT (Liu et al., 2019)	86.42 (↓ 1.51)
KT-NET (Yang et al., 2019)	86.28 (↓ 1.65)

Table 3: Different strategies of introducing knowledge on ARSC dataset.

5 Conclusion

Inspired by human intelligence, we introduce external knowledge into few-shot learning. A parameter generator network is investigated to this end, which can use external knowledge to generate relation network parameters. With these parameters, the relation network can handle diverse tasks with diverse metric. Through various experiments, we demonstrate the effectiveness of our model.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. We also thank Yushan Xie and Zhixing Tian for helpful suggestions.

This work is supported by the National Key RD Program of China (Grant No. 2020AAA0106400), the National Natural Science Foundation of China (Grant No. 61976211 and Grant No. 61806201) and the Key Research Program of the Chinese Academy of Sciences (Grant No. ZDBS-SSW-JSC006).

References

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2019. Few-shot text classification with distributional signatures. *arXiv preprint arXiv:1908.06039*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*.

[3]Knowledge is used to enhance the sentence representation.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Christian Charras and Thierry Lecroq. 2004. Handbook of exact string matching algorithms. In *Citeseer*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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

Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*.

Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. 2016. Dynamic filter networks. In *Advances in Neural Information Processing Systems*.

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

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*.

Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. 2019. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019. K-bert: Enabling language representation with knowledge graph. *arXiv*, pages arXiv–1909.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*.

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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. 2020. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.