Knowledge-Aware Meta-learning for Low-Resource Text Classification

Huaxiu Yao[1†], **Yingxin Wu**[2‡], **Maruan Al-Shedivat**[4§], **Eric P. Xing**[3,4§]
[1]Stanford University; [2]University of Science and Technology of China
[3]Mohamed bin Zayed University of Artificial Intelligence, [4]Carnegie Mellon University
[†]huaxiu@cs.stanford.edu, [‡]wuyxinsh@gmail.com
[§]{alshedivat, epxing}@cs.cmu.edu

Abstract

Meta-learning has achieved great success in leveraging the historical learned knowledge to facilitate the learning process of the new task. However, merely learning the knowledge from the historical tasks, adopted by current meta-learning algorithms, may not generalize well to testing tasks when they are not well-supported by training tasks. This paper studies a low-resource text classification problem and bridges the gap between meta-training and meta-testing tasks by leveraging the external knowledge bases. Specifically, we propose KGML to introduce additional representation for each sentence learned from the extracted sentence-specific knowledge graph. The extensive experiments on three datasets demonstrate the effectiveness of KGML under both supervised adaptation and unsupervised adaptation settings.

1 Introduction

Learning-to-learn (or meta-learning) (Bengio et al., 1990; Schmidhuber, 1992; Hochreiter et al., 2001; Vinyals et al., 2016; Finn et al., 2017) has recently emerged as a successful technique for training models on large collections of low-resource tasks. In the natural language domain, it has been used to improve machine translation (Gu et al., 2018), semantic parsing (Sun et al., 2020), text classification (Bao et al., 2019; Geng et al., 2020, 2019; Li et al., 2020), sequence labelling (Li et al., 2021), text generation (Guo et al., 2020), knowledge graph reasoning (Wang et al., 2019), among many other applications in low-resource settings.

Meta-learning has been shown to dominate self-supervised pretraining techniques such as masked language modeling (Devlin et al., 2018) when the training tasks are representative enough of the tasks encountered at test time (Bansal et al., 2019, 2020). However, in practice, it requires access to a very large number of training tasks (Al-Shedivat et al., 2021) and, especially in the natural language domain, mitigating discrepancy between training and test tasks becomes non-trivial due to new concepts or entities that can be present at test time only.

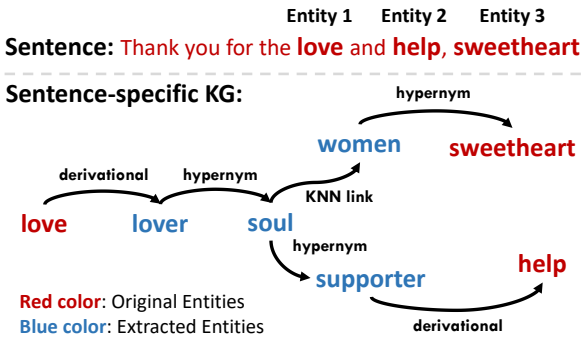


Figure 1: Illustration of extracting a sentence-specific KG from a shared KB.

In this paper, we propose to leverage external knowledge bases (KBs) in order to bridge the gap between the training and test tasks and enable more efficient meta-learning for low-resource text classification. Our key idea is based on computing additional representations for each sentence by constructing and embedding sentence-specific knowledge graphs (KGs) of entities extracted from a knowledge base shared across all tasks (e.g., Fig. 1). These representations are computed using a graph neural network (GNN) which is meta-trained end-to-end jointly with the text classification model. Our approach is compatible with both supervised and unsupervised adaptation of predictive models.

Related work. In modern meta-learning, there are two broad categories of methods: (i) gradient-based (Finn et al., 2017; Nichol and Schulman, 2018; Li et al., 2017; Zhang et al., 2020; Zintgraf et al., 2019; Lee and Choi, 2018; Yao et al., 2019, 2020) and (ii) metric-based (Vinyals et al., 2016; Snell et al., 2017; Yang et al., 2018; Yoon et al., 2019; Liu et al., 2019; Sung et al., 2018). The first category of methods represents the "meta-knowledge" (i.e., a transferable knowledge shared across all tasks) in the form of an initialization of

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1814–1821
November 7–11, 2021.

the base predictive model. Methods in the second category represent meta-knowledge in the form of a shared embedding function that allows to construct accurate non-parametric predictors for each task from just a few examples. Both classes of methods have been applied to NLP tasks (e.g., Han et al., 2018; Bansal et al., 2019; Gao et al., 2019), however, methods that can systematically leverage external knowledge sources typically available in many practical settings are only starting to emerge and focusing on limited applicable scopes (e.g., (Qu et al., 2020; Seo et al., 2020)).

Contributions.

1. We investigate a new meta-learning setting where few-shot tasks are complemented with access to a shared knowledge base (KB).
2. We develop a new method (KGML) that can leverage an external KB and bridge the gap between the training and test tasks.
3. Our empirical study on three text classification datasets (Amazon Reviews, Huffpost, Twitter) demonstrates the effectiveness of our approach.

2 Preliminaries

We consider the standard meta-learning setting, where given a set of training tasks $\mathcal{T}_1, \ldots, \mathcal{T}_n$, we would like to learn a good parameter initialization $\theta_\star$ for a predictive model f_θ such that it can be quickly adapted to new tasks given only a limited amount of data (i.e., few-shot regime). Each task $\mathcal{T}_i$ has a support set of labeled or unlabeled sentences $\mathcal{D}_i^s = \{\mathbf{X}_i^s, \mathbf{Y}_i^s\} = \{(\mathbf{x}_{i,j}^s, \mathbf{y}_{i,j}^s)\}_{j=1}^{N^s}$ and a query set, $\mathcal{D}_i^q = \{\mathbf{X}_i^q, \mathbf{Y}_i^q\} = \{(\mathbf{x}_{i,j}^q, \mathbf{y}_{i,j}^q)\}_{j=1}^{N^q}$ of labeled sentences.

In our text classification setup, we assume that parameters θ are split into two subsets: (1) BERT (Devlin et al., 2018) parameters θ^B shared across tasks and (2) task-specific parameters θ^c that are adapted for each task. Below, we discuss two adaptation strategies: *supervised* and *unsupervised*.

2.1 Supervised adaptation

Under supervised adaptation scenario, we incorporate knowledge with both gradient-based meta-learning and metric-based meta-learning, which are detailed as:

Gradient-based meta-learning. Following Finn et al. (2017), the task-specific parameters θ_i^c for each task $\mathcal{T}_i$ can be adapted by finetuning them on the support set: $\theta_i^c = \theta^c - \alpha \nabla_{\theta^c} \mathcal{L}(f_{\theta^c,\theta^B}; \mathcal{D}_i^s)$, where $\mathcal{L}$ is the cross-entropy loss. Then, using the query set $\mathcal{D}_i^q$, we can evaluate the post-finetuning model and optimize the model initialization as follows:

$$\theta_\star^c, \theta_\star^B \leftarrow \arg\min_{\theta^c,\theta^B} \frac{1}{n} \sum_i \mathcal{L}(f_{\theta_i^c,\theta^B}; \mathcal{D}_i^q) \quad (1)$$

At evaluation time, the initialization parameters $\theta_\star$ are adapted to test tasks $\mathcal{T}_t$ by finetuning on the corresponding support sets $\mathcal{D}_t^s$.

Metric-based Meta-learning. Following (Snell et al., 2017) Prototypical Network (ProtoNet), the task-specific parameter θ_i^c is formulated as a lazy classifier, which is built upon the prototypes $\mathbf{c}_i^k = \frac{1}{|\mathcal{D}_{i,k}^s|} \sum_j f_{\theta^B}(\mathbf{x}_{i,j;k}^s)$. Here, $\mathcal{D}_{i,k}^s$ represents the subset of support sentences belonging to class k. Then, for each sentence in the query set, the probability of assigning it to class k is calculated as:

$$p(\mathbf{y}_{i,j}^q = k|\mathbf{x}_{i,j}^q) = \frac{\exp(-d(f_{\theta^B}(\mathbf{x}_{i,j}^q), \mathbf{c}_i^k))}{\sum_{k'} \exp(-d(f_{\theta^B}(\mathbf{x}_{i,j}^q), \mathbf{c}_i^{k'}))}, \quad (2)$$

where d is defined as a distance measure. During the meta-training phase, ProtoNet learns a well-generalized embedding function $\theta_\star^B$. Then, the meta-learned $\theta_\star^B$ is applied to the meta-testing task, where each query sentence is assigned to the nearest class with the highest probability (i.e., $\hat{\mathbf{y}}_{t,j}^q = \arg\max_r p(\mathbf{y}_{t,j}^q = r|\mathbf{x}_{t,j}^q)$).

2.2 Unsupervised adaptation

When labeled supports sets $\mathcal{D}_i^s$ are not available, we follow Zhang et al. (2020) and use ARM-CML. For each task $\mathcal{T}_i$, we use the shared BERT encoder to compute a representation of each query sentence $\mathbf{x}_{i,j}^q$, which returns an embedding vector, denoted $f_{\theta^B}(\mathbf{x}_{i,j}^q)$. Then, we compute the overall representation of the task by averaging these embedding vectors, $\mathbf{c}_i = \frac{1}{N^q} \sum_{j=1}^{N^q} f_{\theta^B}(\mathbf{x}_{i,j}^q)$. This task representation is then used as an additional input to the sentence classifier, which is trained end-to-end. The meta-training process can be formally defined as:

$$\theta_\star^B, \theta_\star^c \leftarrow \min_{\theta^B,\theta^c} \frac{1}{n} \sum_i \mathcal{L}\left(f_{\theta^B,\theta^c}; \mathcal{D}_i^q, \mathbf{c}_i\right) \quad (3)$$

Note that to enable unsupervised adaptation, ARM-CML learns to compute accurate task embeddings $\mathbf{c}_i$ from unlabeled data instead of using finetuning.

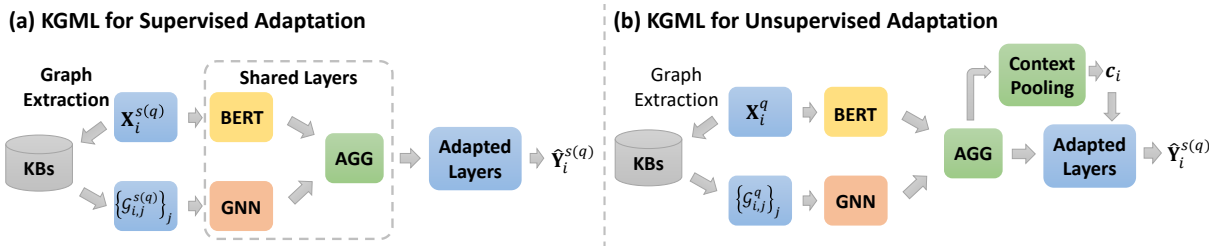


Figure 2: KGML framework on (a) supervised and (b) unsupervised adaptation settings. AGG represents the aggregator AGG_{kf} for knowledge fusion.

3 Approach

In this section, we present the proposed KGML framework (Fig. 2), which allows us to enhance supervised and unsupevised adaptation methods described in the previous section with external knowledge extracted from a shared KB and. In the following subsections, we elaborate the key components of KGML: (1) extraction and representation of sentence-specific knowledge graphs (KGs) and (2) knowledge fusion.

3.1 KG Extraction and Representation

For each sentence $\mathbf{x}_{i,j}$, we propose to extract a KG, denoted $\mathcal{G}_{i,j} = \{\mathcal{N}_{i,j}, \mathcal{E}_{i,j}\}$. The nodes $\mathcal{N}_{i,j}$ of the graph correspond to entities in the corresponding sentence $\mathbf{x}_{i,j}$ and the edges $\mathcal{E}_{i,j}$ correspond to relations between these entities. The relations between the entities are extracted from the KB shared across all tasks. Notice that some entities are not directly related to each other in the KB. To enhance the density of graphs, we further "densify" the extracted KG with additional edges by constructing a k-nearest neighbor graph (k-NNG) based on the node embeddings. More details of KG construction algorithm are provided in Appendix A.

To compute representations of the sentence-specific KGs, we use graph neural networks (GNN) (Kipf and Welling, 2016; Zonghan Wu, 2019). In particular, we use GraphSAGE (Hamilton et al., 2017) as the forward propagation algorithm, which is formulated as follows:

$$\begin{aligned} \mathbf{h}_v^k &= \sigma\left(\mathbf{W}_1^k \cdot \mathbf{h}_v^{k-1} + \mathbf{W}_2^k \cdot \mathbf{h}_{\mathcal{N}(v)}^k\right) \\ \text{s.t. } \mathbf{h}_{\mathcal{N}(v)}^k &= \text{AGG}_k\left(\left\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\right\}\right) \end{aligned} \tag{4}$$

where $\mathbf{W}^k (\forall k \in \{1, ..., K\})$ are the weight matrices of the GNN, $\mathcal{N}(v)$ represents neighborhood set of node v and $\mathbf{h}_u^k$ denotes the node representation in the k-th convolutional layer ($\mathbf{h}_v^0$ as the input feature). σ and AGG_k are functions of non-linearity and aggregator, respectively.

After passing each graph $\mathcal{G}_{i,j}$ into the graph neural network, we aggregate all node representations $\{\mathbf{h}_v^K \mid v \in \mathcal{N}_{i,j}\}$ and output the graph embedding $\mathbf{g}_{i,j}$ as the holistic representation of the knowledge graph.

Algorithm 1 KGML for Supervised Adaptation

Require: Task distribution $p(\mathcal{T})$; Stepsize α, β; Knowledge Base

1: Randomly initialize parameter θ_0, ϕ
2: **while** not converge **do**
3: Sample tasks $\{\mathcal{T}_i\}_{i=1}^{|I|}$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample support set $\mathcal{D}_i^s$ and query set $\mathcal{D}_i^q$
6: Learn the sentence embeddings $f_{\theta^B}(\mathbf{x}_{i,j}^{s(q)})$
7: Extract the knowledge graph $\mathcal{G}_{i,j}^{s(q)}$ for each sentence $\mathbf{x}_{i,j}^{s(q)}$
8: For each graph, using GNN to learn the graph embedding $\mathbf{g}_{i,j}^{s(q)}$ via Eqn. (3)
9: Fuse the sentence and graph embeddings via Eqn. (4) and obtain $\{\tilde{f}_{\theta^B}(\mathbf{x}_{i,j}^{s(q)})\}_{j=1}^{N_{s(q)}}$
10: Compute the task specific parameter θ_i^c for MAML or compute the prototypes $\{\mathbf{c}_i^k\}_{k=1}^K$ for ProtoNet
11: Compute loss $\mathcal{L}(f_{\theta_i^c}(\{\tilde{f}_{\theta^B}(\mathbf{x}_{i,j}^q)\}_{j=1}^{N_q}), \mathbf{Y}_i^q)$
12: **end for**
13: Update all parameters $\theta^c, \theta^B, \phi := \arg\min_{\theta_0,\theta_B,\phi} \frac{1}{|I|}\sum_i \mathcal{L}(f_{\theta_i^c}(\{\tilde{f}_{\theta^B}(\mathbf{x}_{i,j}^q)\}_{j=1}^{N_q}), \mathbf{Y}_i^q)$
14: **end while**

3.2 Knowledge Fusion

To bridge the distribution gap between meta-training and meta-testing stages, we integrate the information extracted from knowledge graph into the meta-learning framework. Assume the sentence representation is $f_{\theta^B}(\mathbf{x}_{i,j})$. For each sentence, we are motivated to design another aggregator AGG_{kf} to aggregate the information captured from the representation of sentence $f_{\theta^B}(\mathbf{x}_{i,j})$ and its corresponding knowledge graph representation $\mathbf{g}_{i,j}$.

Specifically, the aggregator is formulated as:

$$\tilde{f}_{\theta^B}(\mathbf{x}_{i,j}) = \mathrm{AGG}_{kf}(f_{\theta^B}(\mathbf{x}_{i,j}), \mathbf{g}_{i,j}) \quad (5)$$

There are various selections of aggregators (e.g., fully connected layers, recurrent neural network), and we will detail the selection of aggregators in the Appendix D. Then, we replace the sentence representation $f_{\theta^B}(\mathbf{x}_{i,j})$ by $\tilde{f}_{\theta^B}(\mathbf{x}_{i,j})$ in the meta-learning framework. We denote all parameters related to knowledge graph extraction and knowledge fusion as ϕ. Notice that ϕ are globally shared across all task in MAML since we are suppose to connect the knowledge among them. In Alg. 1 and Alg. 3 (Appendix B), we show the meta-learning procedure of the proposed model under the settings of supervised and unsupervised adaption, respectively.

4 Experiments

In this section, we show the effectiveness of our proposed KGML on three datasets and conduct related analytic study.

4.1 Dataset Description

Under the supervised adaptation, we leverage two text classification datasets. The first one is Amazon Review (Ni et al., 2019), aiming to classify the category of each review. The second one is a headline category classification dataset – Huffpost (Misra, 2018), aiming to classify the headlines of News. We apply the traditional N-way K-shot few-shot learning setting (Finn et al., 2017) on these datasets (N=5 in both Huffpost and Amazon Review).

As for the unsupervised adaptation, similar to the settings in (Zhang et al., 2020), we use a federated sentiment classification dataset – Twitter (Caldas et al., 2018), to evaluate the performance of KGML. Each tasks in Twitter represents the sentences of one user. Detailed data descriptions are shown in Appendix C.

4.2 Experimental Settings

For supervised adaptation, we compare KGML on five recent meta-learning algorithms, including MAML (Finn et al., 2017), ProtoNet (Snell et al., 2017), Matching Network (Vinyals et al., 2016) (MatchingNet), REGRAB (Qu et al., 2020), Induction Network (InductNet) (Geng et al., 2019). We conduct the experiments under 1-shot and 5-shot settings and report the results of KGML with gradient-based meta-learning (KGML-MAML) and metric-based meta-learning (KGML-ProtoNet) algorithms.

Under the unsupervised adaptation scenario, KGML is compared with the following four baselines: empirical risk minimization (ERM), up-weighting (UW), domain adversarial neural network (DANN) (Ganin and Lempitsky, 2015), and adaptive risk minimization (ARM) (Zhang et al., 2020). Here, we report the performance with full users and 60% users for meta-training.

On both scenarios, accuracy is used as the evaluation metric and all baselines use ALBERT (Lan et al., 2019) as encoder. WordNet (Miller, 1995) is used as the knowledge graph. All other hyperparameters are reported in Appendix D.

4.3 Overall Performance

The overall performance of all baselines and KGML are reported in Table 1. The results indicate that KGML achieves the best performance in all scenarios by using knowledge bases to bridge the gap between the meta-training and meta-testing tasks. Additionally, under the supervised adaptation scenario, the improvements of Amazon Review are larger than that in Huffpost under the 1-shot setting, indicating that the former has a larger gap between meta-training and meta-testing tasks. One potential reason is that the number of entities of Amazon review is more than Huffpost headlines, resulting in more comprehensive knowledge graphs. Another interesting finding is that ARM hurts the performance under the unsupervised adaptation. However, with the help of the knowledge graph, KGML achieves the best performance, corroborating its effectiveness in learning more transferable representations and further enabling efficient unsupervised adaptation.

4.4 Ablation Study

We conduct ablation studies to investigate the contribution of each component in KGML. Two ablation models are proposed: I. replacing the aggregator AGG_{kf} with a simple feature concatenator; II. removing extra edges in KG, which are introduced by k-nearest neighbor graph. The performance of each ablation model and the KGML of Amazon and Huffpost are reported in Table 2. We observe that (1) KGML outperforms model I, demonstrating the effectiveness of the designed aggregator; (2) Comparing between KGML with model II, the results show that KNN boosts performance. One potential reason is that KNN densifies the whole

Table 1: Performance for supervised and unsupervised adaptation methods. We report the averaged accuracy over 600 tasks (supervised adaptation)/all meta-testing users (unsupervised adaptation).

Supervised Adaptation					Unsupervised Adaptation		
Data	Amazon Review		Huffpost		Data	Twitter	
Shot	1-shot	5-shot	1-shot	5-shot	User Ratio	0.6	1.0
MAML	44.35%	56.94%	39.95%	51.74%	ERM	62.91%	66.05%
ProtoNet	55.32%	73.30%	41.72%	57.53%	UW	63.51%	64.13%
InductNet	45.35%	56.73%	41.35%	55.96%	ARM	60.42%	60.42%
MatchingNet	51.16%	69.89%	41.18%	54.41%	DRNN	63.02%	64.02%
REGRAB	55.07%	72.53%	42.17%	57.66%	-	-	-
KGML-MAML	51.44%	58.81%	**44.29%**	54.16%	**KGML**	**64.92%**	**67.00%**
KGML-ProtoNet	**58.62%**	**74.55%**	42.37%	**58.75%**	-	-	-

network according to the entities' semantic embeddings learned from the original WordNet, which explicitly enriches the semantic information of the neighbor set of each entity. It further benefits the representation learning process and improves the performance.

Table 2: Ablation study (1-shot scenario). Backbone: base meta-learning algorithm

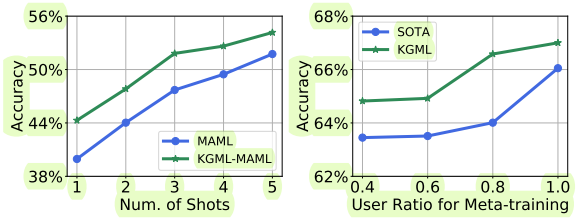
Ablations	Backbone	Amazon	Huffpost
I. Remove AGG_{kf}	MAML	45.68%	41.55%
	ProtoNet	57.94%	41.71%
II. Remove KNN	MAML	51.07%	41.20%
	ProtoNet	57.80%	41.91%
KGML	MAML	51.44%	**44.29%**
KGML	ProtoNet	**58.62%**	42.37%

4.5 Robustness Analysis

In this subsection, we analyze the robustness of KGML under different settings. Specifically, under supervised adaptation, we change the number of shots in Huffpost. Under unsupervised adaptation, we reduce the number of training users in Twitter. The performance are illustrated in Figure 3a and Figure 3b, respectively (see the comparison between Huffpost-ProtoNet and ProtoNet in Appendix E). From these figures, we observe that KGML consistently improves the performance in all settings, verifying its effectiveness to improve the generalization ability.

4.6 Discussion of Computational Complexity

We further conduct the analysis of computational complexity and reported the meta-training time per task in Table 3, where the results of supervised adaptation are performed under the setting of Huffpost 5-shot. Though KGML increases the meta-training time to some extent, the who training process can be finished within 1-2 hours. Thus, the additional computational cost seems to be a reasonable trade-off for accuracy.



(a) Huffpost (b) Twitter

Figure 3: Robustness analysis. SOTA: best baseline

Table 3: Results of meta-training time per task.

Model	Supervised (MAML)	Unsupervised
w/o KG	0.297s	0.146s
with KG	0.407s	0.181s

5 Conclusion

In this paper, we investigated the problem of meta-learning on low-resource text classification, and propose a new method KGML. Specifically, by learning the representation from extracted sentence-specific knowledge graphs, KGML bridges the gap between meta-training and meta-testing tasks, which further improves the generalization ability of meta-learning. KGML is compatible with supervised and unsupervised adaptation and the empirical experiments on three datasets demonstrate its effectiveness over state-of-the-art methods.

Acknowledgments

This work is partially supported by NSF awards IIS-#1617583, IIS-#1955532, CNS-#2008248 and NGA HM-#04762010002. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Maruan Al-Shedivat, Liam Li, Eric Xing, and Ameet Talwalkar. 2021. On data efficiency of meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1369–1377. PMLR.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2019. Learning to few-shot learn across diverse natural language classification tasks. *arXiv preprint arXiv:1911.03863*.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. *arXiv preprint arXiv:2009.08445*.

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

Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. 1990. *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche opérationnelle.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel 2.0: Towards more challenging few-shot relation classification. *arXiv preprint arXiv:1910.07124*.

Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. *arXiv preprint arXiv:2005.05727*.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. *arXiv preprint arXiv:1902.10482*.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. 2018. Meta-learning for low-resource neural machine translation. In *EMNLP*.

Daya Guo, Akari Asai, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Jian Yin, and Ming Zhou. 2020. Inferential text generation with multiple knowledge sources and meta-learning. *arXiv preprint arXiv:2004.03070*.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034.

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.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. 2001. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer.

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

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yoonho Lee and Seungjin Choi. 2018. Gradient-based meta-learning with learned layerwise metric and subspace. In *ICML*, pages 2933–2942.

Zheng Li, Mukul Kumar, William Headden, Bing Yin, Ying Wei, Yu Zhang, and Qiang Yang. 2020. Learn to cross-lingual transfer with meta graph learning across heterogeneous languages. In *EMNLP*, pages 2290–2301.

Zheng Li, Danqing Zhang, Tianyu Cao, Yiwei Song, and Bing Yin. 2021. Metats: Meta teacher-student network for multilingual sequence labeling with minimal supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. 2017. Meta-sgd: Learning to learn quickly for few shot learning. *arXiv preprint arXiv:1707.09835*.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. 2019. Few-shot unsupervised image-to-image translation. In *ICCV*, pages 10551–10560.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Rishabh Misra. 2018. News category dataset.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197.

Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*.

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *International Conference on Machine Learning*, pages 7867–7876. PMLR.

Jürgen Schmidhuber. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Learning*, 4(1).

Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. 2020. Physics-aware spatiotemporal modules with auxiliary tasks for meta-learning. *arXiv preprint arXiv:2006.08831*.

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

Yibo Sun, Duyu Tang, Nan Duan, Yeyun Gong, Xiaocheng Feng, Bing Qin, and Daxin Jiang. 2020. Neural semantic parsing in low-resource settings with back-translation and meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8960–8967.

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, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *NIPS*, pages 3630–3638.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Meta reasoning over knowledge graphs. *arXiv preprint arXiv:1908.04877*.

Wikipedia. 2021. Minimum spanning tree — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Minimum%20spanning%20tree&oldid=1015496878. [Online; accessed 17-May-2021].

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*.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. 2019. Hierarchically structured meta-learning. In *ICML*.

Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. 2020. Automated relational meta-learning. In *ICLR*.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*.

Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. 2020. Adaptive risk minimization: A meta-learning approach for tackling group distribution shift. *arXiv preprint arXiv:2007.02931*.

Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. 2019. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR.

Fengwen Chen Guodong Long Chengqi Zhang Zonghan Wu, Shirui Pan. 2019. A comprehensive survey on graph neural networks.

A Detailed Descriptions of Sentence-specific KG Construction

To construct a holistic knowledge graph with all the entities and relations, we first use the existing knowledge graph (i.e., WordNet (Miller, 1995)) as the sparse knowledge base $\mathcal{G}^{base}$. Some entities may be the nodes with few interactions or even isolated. Thus, to connect all entities, the node embeddings in the knowledge base are then used to construct a K-NN graph $\mathcal{G}^{knn}$, which is further combined with the base knowledge graph, rendering the dense knowledge graph $\mathcal{G} = \mathcal{G}^{knn} \cup \mathcal{G}^{base}$.

For each sentence $\mathbf{x}_{i,j}$, we use its entities to query the knowledge graph $\mathcal{G}$, which returns the entity embeddings $\mathcal{N}_{i,j}$ and a adjacency matrix $\mathcal{A}_{i,j}$. Each element in $\mathcal{A}_{i,j}$ represents the shortest distance of the corresponding entities. Inspired by Occam's Razor criterion, we compute the Minimum Spanning Tree (MST) (Wikipedia, 2021) w.r.t all the target entities (other entities and relations in the chosen path are included) as the concise and informative graphical representation of the sentence. In Alg. 2, we illustrate the whole process of the knowledge graph.

B Pseudocodes of KGML

In this section, we add the pseudocode for unsupervised adaptation in Alg. 3.

Algorithm 2 Knowledge Graph Extraction

Require: Dense knowledge graph $\mathcal{G}$

1: **for** each sentence $\mathbf{x}_{i,j}$ **do**
2: Use the entities $\mathbf{x}_{i,j}$ to query $\mathcal{G}$ and obtain entity embeddings $\mathcal{N}_{i,j}$ and adjacency matrix $\mathcal{A}_{i,j}$
3: Apply MST algorithm on $\mathcal{A}_{i,j}$, which returns T, the minimum spanning tree w.r.t the entities in $\mathbf{x}_{i,j}$.
4: Construct the knowledge graph $\mathcal{G}_{i,j}$ by including the selected nodes and edges on the path of T, i.e., $\mathcal{G}_{i,j} = \{(r,s) \mid \exists (u,v) \in \mathrm{T}, (r,s) \in \mathrm{ShortestPath}(u,v)\}$.
5: **end for**

C Data Statistics

For supervised adaptation, we use Amazon Review and Huffpost to evaluate the performance. Amazon Review contains 28 classes, and the number of classes for meta-training, meta-validation, and meta-testing are 15, 5, 8, respectively. The Huffpost dataset includes 41 classes in total, and we use 25, 6, 10 classes for meta-training, meta-validation, and meta-testing, respectively. In terms of the unlabeled adaptation, the number of Twitter users for meta-training, meta-validation, and meta-testing are 741, 92, 94, respectively.

D Hyperparameter Settings

For all the supervised adaptation and the unlabeled adaption experiments, we use ALBERT (Lan et al., 2019) as the sentence encoders and GraphSAGE (Hamilton et al., 2017) as the graph encoders. All hyperparameters are selected via the performance on the validation set.

D.1 Supervised Adaptation

The GNN used contains two layers, where the number of neurons is 64 and 16, respectively. We adopt two fully connected layers with ReLU as activation layer for the adaptation layers, where the number of neurons is 64 for each layer. The aggregator AGG_{kf} is designed as the one fully connected layer. We set the inner-loop learning rate α and outer-loop learning rate β as 0.01 and 2e-5, respectively. The number of steps in the inner loop is set as 5. We use Adam (Kingma and Ba, 2014) for outer loop optimization. The maximum number of epochs for huffpost and Amazon Review is 10,000 and 4,000, respectively.

Algorithm 3 KGML for Unsupervised Adaptation

Require: Task distribution $p(\mathcal{T})$; Stepsize β; Knowledge Base

Randomly initialize parameter θ_0, ϕ
while not converge **do**
Sample tasks $\{\mathcal{T}_i\}_{i=1}^{|I|}$
for all $\mathcal{T}_i$ **do**
Sample query set $\mathcal{D}_i^q$ from the task $\mathcal{T}_i$
Learn the sentence embeddings $f_{\theta^B}(\mathbf{x}_{i,j}^q)$
Extract the knowledge graph $\mathcal{G}_{i,j}^q$
For each graph, using GNN to learn the graph embedding $\mathbf{g}_{i,j}^q$ via Eqn. (3)
Fuse the sentence and graph embeddings and obtain the final embedding $\{f_{\theta^B}(\mathbf{x}_{i,j}^q)\}_{j=1}^{N_s}$
Calculate the contextual vector $\mathbf{c}_i$ and compute loss $\mathcal{L}(f_{\theta_0}(\{f_{\theta^B}(\mathbf{x}_{i,j}^q)\}_{j=1}^{N_q}, \mathbf{c}_i), \mathbf{Y}_i^q)$
end for
Update all parameters $\theta^c, \theta^B, \phi := \arg\min \frac{1}{|I|} \sum_i \mathcal{L}(f_{\theta^c}(\{\tilde{f}_{\theta^B}(\mathbf{x}_{i,j}^q)\}_{j=1}^{N_q}, \mathbf{c}_i), \mathbf{Y}_i^q)$
end while

D.2 Unsupervised Adaptation

For the sentence encoder, the number of output dimensions is set as 240. The GNN is composed of two convolution layers, where each layer contains 64 neurons, and AGG_k is designed as a mean pool operation. We use one fully connected layer for the final aggregation AGG_{kf}. In the training phase, the learning rate is set β as 1e-4, and we use Adam (Kingma and Ba, 2014) optimizer with weight decay 1e-5. The contextual support size and meta batch size are 50 and 2, respectively.

E Additional Results of Robustness Analysis

In Figure 4, we show the comparison between Huffpost-ProtoNet and ProtoNet w.r.t. the number of shots. The results further demonstrate the effectiveness of KGML.

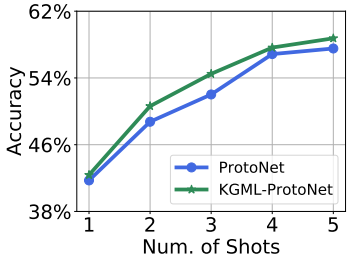


Figure 4: Additional Robustness analysis.