big data and cognitive computing

MDPI

Article

Knowledge-Enhanced Prompt Learning for Few-Shot Text Classification

Jinshuo Liu and Lu Yang *

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China; liujinshuo@whu.edu.cn
* Correspondence: yanglu93@whu.edu.cn

Abstract: Classification methods based on fine-tuning pre-trained language models often require a large number of labeled samples; therefore, few-shot text classification has attracted considerable attention. Prompt learning is an effective method for addressing few-shot text classification tasks in low-resource settings. The essence of prompt tuning is to insert tokens into the input, thereby converting a text classification task into a masked language modeling problem. However, constructing appropriate prompt templates and verbalizers remains challenging, as manual prompts often require expert knowledge, while auto-constructing prompts is time-consuming. In addition, the extensive knowledge contained in entities and relations should not be ignored. To address these issues, we propose a structured knowledge prompt tuning (SKPT) method, which is a knowledge-enhanced prompt tuning approach. Specifically, SKPT includes three components: prompt template, prompt verbalizer, and training strategies. First, we insert virtual tokens into the prompt template based on open triples to introduce external knowledge. Second, we use an improved knowledgeable verbalizer to expand and filter the label words. Finally, we use structured knowledge constraints during the training phase to optimize the model. Through extensive experiments on few-shot text classification tasks with different settings, the effectiveness of our model has been demonstrated.

Keywords: text classification; prompt learning; knowledge enhancement; few-shot learning

Citation: Liu, J.; Yang, L. Knowledge-Enhanced Prompt Learning for Few-Shot Text Classification. *Big Data Cogn. Comput.* **2024**, *8*, 43. https://doi.org/10.3390/bdcc8040043

Academic Editors: Tim Schlippe and Matthias Wölfel

Received: 17 March 2024
Revised: 11 April 2024
Accepted: 12 April 2024
Published: 18 April 2024

1. Introduction

In recent years, pre-trained language models (PLMs) [1,2] have achieved state-of-the-art results in many benchmarks in the field of natural language processing, such as question answering [3] and text classification [4]. However, for specific downstream tasks, fine-tuning PLMs is still necessary, which is time-consuming, and the generalization ability of the model heavily depends on a large amount of annotated data and computational resources. These problems lead to an important research task, few-shot learning, which can learn new categories through several labeled samples and has good generalization ability. At present, few-shot learning has been extended to a variety of natural language processing tasks, including text classification [5], relation extraction [6], and so on. Furthermore, few-shot text classification has important research significance due to its wide range of application scenarios. The objective of few-shot text classification is to train a classifier when only a few training samples are given for each category.

Since PLMs are trained on a large amount of general corpus, they contain abundant knowledge and can be taught using few training samples [7]. However, the application of the knowledge contained in the model to solve few-shot text classification requires additional fine-tuning. In addition, due to the small number of labeled training samples provided by the few-shot text classification task, it is hard to achieve a good fitting effect, and the model has poor generalization performance [7]. Therefore, prompt learning has become an effective new approach for addressing few-shot learning tasks. From the time that GPT-3 [7] was introduced, prompt learning has worked to bridge the gap between

PLMs and various downstream tasks, since it converts downstream tasks into pre-training tasks for PLMs. Prompt learning is more effective in low-resource settings because there are not enough training samples to specify the behavior of the model; therefore, using text prompts to push the model in the correct direction is particularly effective. One of the typical ways to use prompt learning is to formalize the text classification task into a masked language modeling problem, which can better activate the ability of the masked language model. Additionally, because most of the prompt learning methods usually keep the parameters of a PLM fixed, they require fewer training parameters, leading to reduced training costs.

Although traditional prompt learning methods have addressed the shortcomings of fine-tuning PLMs, there are still other issues: on one hand, it is still hard to construct appropriate prompt templates, because manual prompt templates often require expert knowledge [8], while auto-constructing prompt templates often requires significant computation cost [9]. On the other hand, the input text includes semantic knowledge as well as structured knowledge, which is overlooked by existing prompt learning methods.

To address these issues, we propose SKPT for few-shot text classification. Specifically, SKPT includes three components: prompt template construction, prompt verbalizer construction, and training strategies. First, we extract open entities and relations through open information extraction (OpenIE) [10]. Based on the semantic information and the location of open triples, we propose a structured knowledge template and initialize it. Second, we use an improved knowledgeable verbalizer to expand and filter the label words. Finally, we use structured knowledge constraints during the training phase to optimize the embedding. We conduct experiments on two topic text classification datasets, AG News [11] and DBpedia [12]. The results demonstrate the effectiveness of SKPT, and we show the effectiveness of each component through an ablation study.

The main contributions of this paper are as follows:

- We propose a knowledge-enhanced prompt learning method SKPT for few-shot text classification. Based on open triples, we insert learnable virtual tokens into the prompt template to introduce external knowledge.
- We use an improved knowledgeable verbalizer, which utilizes external knowledge bases to expand each class label into a set of label words. We filter the out-of-vocabulary words for the PLM and assign learnable weights to different label words for training.
- We apply structured knowledge constraints during the training phase through the specific loss function.
- We perform experiments on two text classification benchmark datasets to illustrate the effectiveness of SKPT, especially in low-resource settings.

2. Related Works

Our research goal is to enhance prompt learning with the knowledge to address the few-shot text classification problem. Therefore, we focus on the existing research on both prompt learning methods and knowledge-enhanced methods.

2.1. Prompt-Tuning

Many large-scale pre-trained models have been open-sourced in the current research domain, but fine-tuning is still required to adapt PLMs to target downstream tasks. However, there are still some issues in fine-tuning. On one hand, fine-tuning usually requires additional network structures for different downstream tasks; for example, we need to add a classifier for text classification tasks. Therefore, fine-tuning has weak generalization ability for different tasks. On the other hand, the amount of parameters that are required to be trained during the fine-tuning process is still considerable.

Since the introduction of GPT-3 [7], prompt-based learning has attracted significant attention and has achieved remarkable performance in a variety of natural language processing tasks. By using prompt information, we can formalize a downstream task into a pre-training task of a pre-trained model. Since then, numerous effective approaches

have been developed for prompt learning [13,14]. Schick et al. [15] first propose a prompt tuning method based on the manual prompt template, which obtains excellent results on few-shot classification tasks. Prompt learning methods can be categorized into two types: cloze prompts [16], which insert a slot in the middle of the input text separately; and prefix prompts [17,18], where the prompt template token comes before the input text. Due to the significant time and experience required for manually designing templates, and the fact that even experienced prompt designers are unable to manually discover the best prompt information [19], some work has begun to explore automatic search protocols for prompt templates and verbalizers.

Shin et al. [9] propose a gradient-guided search to automatically generate templates and label words. Gao Tianyu et al. [20] propose a prompt-based method with automatically searched prompts, which can also select task demonstrations in the context of the input. However, the quality of prompts from an automatic search in a discrete space is often not the best, and automatic search requires a large amount of computing resources.

Recently, some external knowledge-enhanced prompt learning methods have also been proposed. Liu et at. [21] use external knowledge to design knowledge-enhanced prompts. KP4SR [22] is proposed to use structured knowledge to generate prompts for sequential recommendation tasks. Hu et al. [23] propose a knowledge-enhanced prompt method named KPT, which expands label words through external knowledge bases. Since the key to prompt learning is to design an appropriate prompt template and prompt verbalizer based on the verbalizer of KPT, we propose a new knowledge-enhanced prompt learning method for few-shot text classification. We integrate knowledge into the prompt template as well as the prompt verbalizer to optimize prompt learning.

2.2. Knowledge-Enhanced Methods

The foundation for incorporating knowledge into other PLMs or prompt-based methods is knowledge representation. Knowledge graph embedding serves as a technological application of knowledge representation. Knowledge graph embedding is originally introduced to solve the knowledge graph completion problem. It projects the entities and relations of the knowledge graph to a vector space, thereby preserving the structured information in the knowledge graph. Knowledge graph embedding methods can be mainly divided into two categories: distance-based models such as TransE [24], and semantic matching models such as RESCAL [25] and DistMult [26]. Among them, distance-based models mainly include translation models and rotation models. The representatives of translation models are TransE, TransH [27], and TransR [28], and they consider the relation in a triple as a translation from the head entity to the tail entity. Rotation models consider the relation in a triple as a rotation from the head entity to the tail entity, such as RotatE [29] and QuatE [30]. Semantic matching models obtain the semantic similarity by using the bilinear product [25] or neural network [31]. In our model, we optimize the embedding representation based on the idea of translation models.

Although the research methods for knowledge-enhanced prompt learning are not numerous, there is a considerable amount of research on knowledge-enhanced pre-trained language models [32–34]. The simplest knowledge injection method is just combining entity embedding with the word embedding of pre-trained models [35], but the problem with this method is that embedding obtained from multiple sources cannot be aligned. Meanwhile, in the process of introducing knowledge, there is a risk of also introducing noise. To solve this problem, CoLAKE [36] treats words in the text as nodes in a fully connected graph, identifies entities in them, and links triple knowledge, achieving a 5.6% improvement over BERT in entity classification tasks. K-BERT [37] introduces a method that combines external knowledge with pre-trained models, utilizing sentence trees and a visibility matrix to avoid knowledge noise. Knowledge-enhanced pre-trained language models are of great importance for the development of knowledge-enhanced prompt learning methods, because although knowledge-enhanced pre-trained language models incorporate knowledge during the pre-training phase, and knowledge-enhanced prompt learning methods introduce knowledge

during the prompt tuning phase, the core idea behind both approaches is to enhance the power of pre-trained models with external knowledge for downstream tasks.

3. Methods

The model presented in this paper, named SKPT, consists of three main components: knowledge-enhanced prompt template, knowledge-enhanced verbalizer, and training strategies. The model is illustrated in Figure 1. First, we extract open entities and relations from text data through open information extraction (OpenIE) [10] and propose a structured knowledge prompt template based on these triples, which initializes the template embedding by introducing prior knowledge. Second, we use the improved knowledgable prompt tuning verbalizer to expand class labels into a set of label words with different levels and perspectives and then filter them. On one hand, we filter the out-of-vocabulary (OOV) words for the PLM. On the other hand, we assign learnable weights to different label words for training. Finally, we design a loss function under structured knowledge constraints based on the classic translation model TransE and refine the model parameters through the incorporation of context.

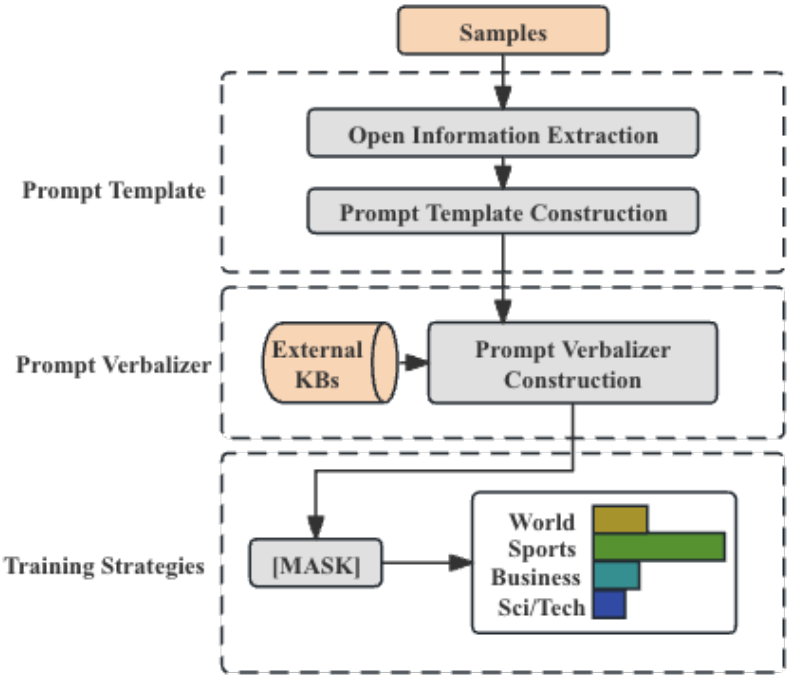

Figure 1. The illustration of SKPT.

3.1. Knowledge-Enhanced Prompt Template

Prompt learning mainly consists of two components, prompt template and prompt verbalizer. The design of prompt templates is crucial for leveraging the model's abilities and has a significant impact on the accuracy of the model. Therefore, the key to improving task performance is to construct prompt templates for specific tasks and use knowledge prompts to guide PLMs.

There are many forms of prompt templates, such as prefix templates, cloze templates, and so on. Different templates are suitable for different downstream tasks, for example, prefix templates are usually more suitable for solving generative tasks. For few-shot text classification tasks, we choose the cloze template as our prompt template. To solve the text classification task, the prompt template formalizes this task as the pre-training task of the masked language model by adding additional template tokens. x represents input data, and the goal of our task is to classify x into a class label $y \in \mathcal{Y}$. Taking a manual template as an example, after being wrapped by the template, the template is as follows:

$$x_{prompt} = [CLS]\ x\ It\ was\ [MASK].\ [SEP] \quad (1)$$

where $[CLS]$and $[SEP]$ are special tokens used to represent the beginning and separation of sentences, respectively.

In this case, the task of text classification is transformed into a cloze task where the $[MASK]$ position needs to be completed. Following this, we use the PLM to encode x_{prompt} and obtain the probability distribution $P([MASK]|x_{prompt})$, which describes the probability of words in $\mathcal{Y}$ being able to replace the [MASK] token.

We propose a new prompt template, a structured knowledge prompt template, which can inject external knowledge from open triples into the prompt template.

Therefore, we first extract entity and relation objects through relation extraction. We use open information extraction methods to perform open relation extraction on the input text, extracting structured knowledge from the data. Open information extraction is a method to extract open triples from unstructured text. It can extract triples from text without a predetermined word set, and the system consists of a series of relation extraction models. Open triples consist of open entities and relations. Different from defined entities and relations, open entities and relations do not have a unified standard but contain more information. Figure 2 shows an example of triple extraction.

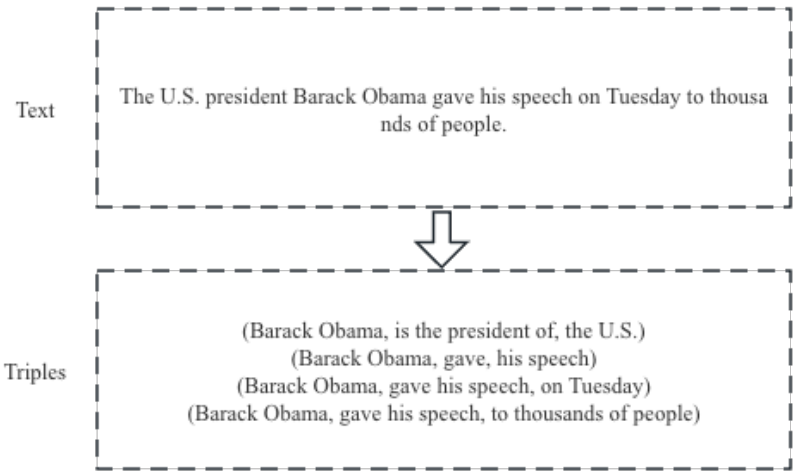

Figure 2. Example of triple extraction. On the top is an unstructured input text; at the bottom are some example triples corresponding to the original text.

We input a sentence x, which includes n tokens, which is denoted as $x = \{v_1, v_2, v_3, \ldots, v_n\}$. After relation extraction, we obtain triples $\{v_h, v_t, v_r\}$. Among them, v_h represents the head entity, v_t represents the tail entity, and v_r represents the relation. Both entities and relations can be composed of multiple tokens, which are simplified by v_h, v_t, and v_r. Through the extraction process, we can obtain two types of useful information: the entities and relations themselves, and the positions of entities and relations in the original input.

We introduce virtual template tokens based on the manual template to incorporate external knowledge. In detail, we insert a token on one side of the located head entity v_h, tail entity v_t, and relation v_r as a part of our prompt template. For instance, when the extracted triple is ("Barack Obama", "gave", "his speech"), we insert the virtual token [entity] before the head entity "Barack Obama" and the tail entity "his speech", as well as the virtual token [relation] before the relation "gave". The use of virtual tokens in the prompt template allows for the introduction of prior knowledge as well as enables the incorporation of structured information between relations and entities during the training process. This is illustrated in Figure 3.

The template tokens we inserted are virtual and need to be initialized first. The process of embedding initialization is the process of introducing prior knowledge. The task is text classification, and open triples, to some extent, contain the main information of the input

text. Therefore, triples also contain the category information of the entire sentence, to a certain extent. For example, in the sentence "Swimming is a very effective exercise.", after performing open information extraction, we know the head entity is "swimming", the tail entity is "exercise" and the relation is "is". Both the head and tail entity point to the class "sports".

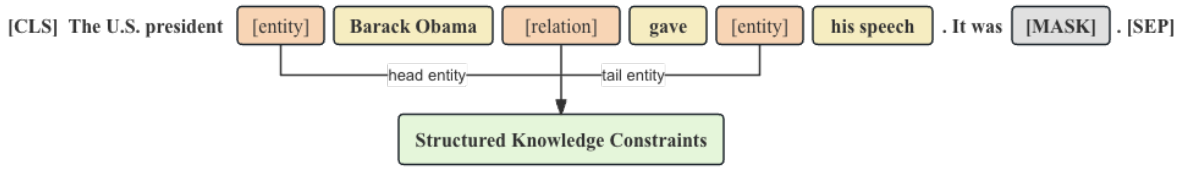

Figure 3. The illustration of structured knowledge prompt template.

For entities, we only need to consider whether their categories are included in the class labels set $\mathcal{Y}$. The distributions of v_h and v_t over $\mathcal{Y}$ are denoted as ϕ_h and ϕ_t, and they are computed based on normalized statistical data. In detail, we encode the texts of entity v_h and all the category texts with RoBERTa, then we compute the cosine similarity for v_h across all categories and normalize these similarities to obtain ϕ_h. The calculation process for ϕ_t is also the same. We use the pre-trained language model to encode the class labels set $\mathcal{Y}$, so we obtain the class labels embedding set $PLM(\mathcal{Y})$. Therefore, we obtain the weighted embedding e_h and e_t of virtual tokens v_h and v_t, and they are initialized as follows:

$$e_h = \phi_h \cdot PLM(\mathcal{Y}) \tag{2}$$

$$e_t = \phi_t \cdot PLM(\mathcal{Y}) \tag{3}$$

3.2. Knowledge-Enhanced Prompt Verbalizer

After processing with the prompt template, prompt learning maps the predicted label words to the real classes through a verbalizer. Let $\mathcal{M} : \mathcal{V} \to \mathcal{Y}$ be a mapping from the label words $\mathcal{V}$ of PLM to the class labels.

In existing research, the prompt verbalizer often maps a label word to each category, whereas it is more realistic for a category to have label words that represent multiple levels and perspectives. We use the verbalizer of KPT [23], which is based on external knowledge bases, to expand label words. Furthermore, we improve the verbalizer.

First, we need to construct a prompt verbalizer, which primarily involves leveraging external knowledge bases to expand the existing class words. We use Related Words [38] as our external knowledge base, which is a knowledge graph that integrates many resources, including WordNet [39] and ConceptNet [40], and is denoted as G. We use the class label v for each category as the source of search and select neighbor nodes of v in G, denoting the set as $N_G(v)$. The correlation score of the neighbor node should be less than a threshold θ; thus, we obtain the set of label words $N_G(v) \cup \{v\}$ corresponding to each category. Therefore, a category's label is expanded into a set of label words that represent different levels and perspectives.

The set of label words obtained by expanding through knowledge bases may not be effective, and they may contain some noise. Therefore, it is necessary to filter the verbalizer.

First, there are some words that are out of vocabulary for the PLM in our expanded set of label words $\mathcal{V}$. Theoretically, out-of-vocabulary words, when processed by a PLM, should result in a probability of 0. However, these words have real meanings; so, in order to avoid ignoring the meaning of these OOV words, we assign probabilities to them. Let an OOV word be denoted as v_o, and let the set of words that are not out of vocabulary be denoted as $\mathcal{V}_{no}$. We can calculate its probability by taking the average probability of all other words that are not OOV from the set $\mathcal{V}_{no}$, that is, we can perform the following calculation:

$$P_{PLM}([MASK] = v_o | x_{prompt}) = \frac{1}{|\mathcal{V}_{no}|} \sum_{v \in \mathcal{V}_{no}} P_{PLM}([MASK] = v | x_{prompt}) \tag{4}$$

Second, in few-shot learning, we can calculate the impact of each label word on the prediction results. Furthermore, the crucial aspect of the verbalizer's filtering algorithm is to retain high-quality label words while removing low-quality ones. Consequently, we provide each label word in label words set $\mathcal{V}_y$ with a w_v, which is a learnable weight in the training phase and is fixed in the testing phase. The predicted label $\hat{y}$ is as follows:

$$\hat{y} = argmax_{y \in \mathcal{Y}} \frac{exp(h(y|x_{prompt}))}{\sum_{y'} exp(h(y'|x_{prompt}))}, \tag{5}$$

where $h(y|x_{prompt})$ is

$$h(y|x_{prompt}) = \sum_{v \in \mathcal{V}_y} w_v log P_{PLM}([MASK = v]|x_{prompt}) \tag{6}$$

3.3. Training Strategies

Pre-trained language models can be divided into various types based on the type of pre-training task. Common examples include left-to-right language models, such as GPT [7], masked language models, like BERT [1], and encoder–decoder language models, like T5 [2]. Different language models are suitable for different downstream tasks. For instance, left-to-right language models, given their computational direction from left to right, are more suitable for prediction tasks. Masked language models are more suited to our few-shot text classification tasks, so RoBERTa [41] is chosen as the main pre-trained model in this paper. After processing with the prompt template, the few-shot text classification task is transformed into a cloze task of RoBERTa, allowing for the better utilization of PLM and bridging the gap between the PLM and the downstream tasks.

Compared to normal prompt learning without knowledge injection, the SKPT template incorporates triple information into the template. Our loss function consists of two parts: a structured knowledge loss function denoted as $Loss_{SK}$ and a cross-entropy loss function for the entire prompt template denoted as $Loss_{MLM}$.

$$Loss = \alpha_1 Loss_{SK} + \alpha_2 Loss_{MLM}, \tag{7}$$

where α_1 and α_2 are weight parameters, and the sum of them is 1.

The entities and relations in triples contain structured knowledge, which is not utilized in normal prompt learning. In addition, after embedding initialization, the virtual template tokens are not associated with contextual information. The principle of the translation model TransE in the knowledge graph, that is, $h + r \approx t$, states that the head entity vector plus the relation vector is approximately equal to the tail entity vector in the vector space. Therefore, we propose a new structured knowledge constraint loss function to associate the virtual template token with its surrounding context.

$$Loss_{SK} = -logsigmoid(\gamma - ||\mathbf{v_h} + \mathbf{v_r} - \mathbf{v_t}||_2^2) - \sum_{i=1}^{n} \frac{1}{n} logsigmoid(||\mathbf{v'_{hi}} + \mathbf{v'_{ri}} - \mathbf{v'_{ti}}||_2^2 - \gamma), \tag{8}$$

where γ represents the margin. The calculation process includes two types of training samples, namely positive samples and negative samples. Positive samples are triples obtained by open information extraction. We randomly shuffle the extracted entities and relations to construct n negative samples, ensuring that the negative samples do not belong to the set of original triples from the dataset.

There are two types of parameters in the model: parameters from the PLM and parameters from prompt learning. We adopt a training strategy that freezes the parameters of the PLM and optimizes only the parameters of the prompt.

4. Experiments

In this section, we conduct a series of experiments with different settings and provide explanations for benchmark datasets and comparative models.

4.1. Datasets

Our model is evaluated using two topic classification benchmark datasets, AG News [11] and DBpedia [12]. AG News: AG News is a topic classification dataset about news, where each sample contains a headline and content body. In this dataset, samples have to be classified as World, Sports, Business, and Science. Each class contains 30,000 training samples and 1900 testing samples. DBpedia: The DBpedia ontology dataset includes fourteen non-overlapping categories selected from DBpedia 2014. Every class contains 40,000 training samples and 5000 testing samples.

We define labeled samples as training set D_{train} and unlabeled samples as testing set D_{test}. We randomly select data from the training set in the form of k-shot and n-ways to construct the few-shot training dataset D_{train}. K-shot represents the number of samples in each category; here, we set k as 5, 10, and 20 in our main experiments. N-ways represents the number of categories in the dataset; for example, n is 14 for the DBpedia dataset, and for the AG News dataset, n is 4. Meanwhile, the number of samples in the validation set D_{val}, and the training set remains the same. The testing set D_{test} is the original testing set of the benchmark dataset.

4.2. Experiment Settings

We use the Micro-F1 score as our test metric, so all experimental results are the F1 scores from the testing phase. For PLM, we utilize $RoBERTa_{large}$ [41] for all experiments to make a fair comparison. Further, we use OpenIE 6 [10] to extract the open triples.

We conduct 5 shot, 10 shot, and 20 shot experiments following KPT [23] to measure the performance across these two datasets. To reduce the error caused by randomness, every experiment uses a fixed seed set, and we take the average value as the final result.

During the training, we set the maximum epochs to 5, the batch size to 2, and the maximum sequence length to 512. To improve the training process, we employ a learning rate warm-up strategy. Specifically, we initialize the learning rate to 3×10^{-4} and set the warm-up steps to 500.

4.3. Baselines

Fine-tuning: Fine-tuning is the further training of PLMs using a few sample training data to adapt to new tasks or domains. The traditional fine-tuning method is to input the embedding of [CLS] into the classification layer for classification.

Prompt-tuning: We choose the typical prompt learning method PET as an example of prompt-tuning. The prompt template of PET [15] uses a manual template. Further, for its verbalizer, there is only one label word corresponding to a category. In order to represent the general prompt-tuning method, we do not include PET's additional optimization method in the experiment.

Knowledgeable Prompt-tuning (KPT) [23]: This is a typical method of knowledge-enhanced prompt-tuning, which focuses on incorporating external knowledge into the prompt verbalizer. Specifically, it utilizes external knowledge bases to introduce external knowledge, expand the label words, and filter the prompt verbalizer. As our model extends and improves some of its methods, we consider KPT as one of the baselines.

5. Results

5.1. Main Results

By comparing our model with the baseline methods through experiments, we demonstrate the effectiveness of SKPT in the few-shot text classification task.

From Table 1, we find that SKPT outperforms fine-tuning, prompt-tuning, and KPT in most of the experiments, especially in the 5 shot and 10 shot experiments, which proves the effectiveness of our method.

Table 1. Results of few-shot text classification.

Shot	Method	AG News	DBpedia
5 shot	Fine-tuning	0.375	0.955
	PET	0.864	0.965
	KPT	0.865	0.968
	SKPT	0.873	0.973
10 shot	Fine-tuning	0.770	0.944
	PET	0.869	0.970
	KPT	0.876	0.979
	SKPT	0.881	0.980
20 shot	Fine-tuning	0.848	0.978
	PET	0.869	0.973
	KPT	0.869	0.979
	SKPT	0.868	0.979
Full dataset	Fine-tuning	0.944	0.991
	SKPT	0.952	0.991

Specifically, our model shows an average improvement of up to 20.9% and 1.93% compared with fine-tuning. In particular, in the AG News dataset, our method is 49.8% superior to fine-tuning. But overall, the gap between SKPT and other methods decreases with the increase in labeled training samples. For 20 shot, we believe that the training samples are enough to optimize the model; so, this model requires little additional external knowledge. Thus, we know that SKPT generalizes well in low-resource settings. For DBpedia, we find that the advantage of SKPT is not obvious. On one hand, it is evident from Table 1 that the DBpedia dataset has already reached a performance level above 0.9, suggesting that there is limited scope for further improvement. On the other hand, DBpedia can be utilized as a knowledge graph, so external knowledge from knowledge graphs could lead to limited effectiveness.

For few-shot learning, the training data typically consists of a few or several dozen examples. To assess the experimental performance of the task under standard training resources, we conduct experiments with fine-tuning and KPT using the entire dataset. From Table 1, we find that both models perform better with the full dataset, but the computational cost is too high in this experimental setting. Furthermore our model also achieves comparable results to the baseline model fine-tuning under the standard training resources.

5.2. Ablation Study

To verify the effectiveness of each component in SKPT, we conduct ablation experiments on the AG News dataset. Considering that in comparative experiments, knowledge injection is more effective under low-resource settings, we believe that 20 training samples are sufficient to provide the textual information for training. So, we only conduct experiments separately under the 5 shot and 10 shot settings. We propose three additional model structures.

To validate the effectiveness of the structured knowledge template, we propose $Model_1$, which differs from SKPT in that it uses a simple manual template. Meanwhile, since there are no triples, $Loss_{SK}$ is not added during the training process, so the model just uses the cross-entropy loss in the training process.

To validate the effectiveness of the knowledge-enhanced verbalizer, we propose $Model_2$. The difference between $Model_2$ and SKPT is the verbalizer. For the verbalizer in $Model_2$, one category is only mapped to one label word.

To validate the effectiveness of the structured knowledge constraint, we propose $Model_3$. The only difference between $Model_3$ and our model is that $Model_3$ only uses cross-entropy loss to optimize the model.

From Table 2, we know that the results of all these three models do not exceed our model. Therefore, all components in our model are necessary. With the use of a normal

prompt template, $Model_1$ achieves less impressive results, suggesting the importance of incorporating knowledge into the template. For most of the results, $Model_2$ achieves better performance among these three comparison models, suggesting that the knowledge verbalizer may not have a significant effect on the datasets.

Table 2. Ablation study of SKPT on AG News.

Shot	Method	AG News	DBpedia
5 shot	$Model_1$	0.859	0.962
	$Model_2$	0.865	0.965
	$Model_3$	0.862	0.963
	SKPT	0.873	0.973
10 shot	$Model_1$	0.875	0.969
	$Model_2$	0.874	0.977
	$Model_3$	0.876	0.976
	SKPT	0.881	0.980

6. Conclusions

In this paper, we present SKPT for few-shot text classification, which enhances knowledge in three phases. First, we insert structured knowledge prompt template tokens into a manual template based on open triples, and these virtual template tokens are initialized by prior knowledge. Second, we use the improved knowledgeable verbalizer to expand the label words based on external knowledge bases. Third, we use structured knowledge constraints during the training phase. We find that SKPT achieves good results in the few-shot text classification task. Our work achieved the highest F1 score compared with baselines, especially in low-resource settings.

However, our research still has certain limitations. We introduce knowledge at three stages: constructing the prompt template, constructing the prompt verbalizer, and training. This leads to an intricate process. In the future, we will extract more effective features from external knowledge graphs or knowledge bases to enhance the performance of PLMs in few-shot text classification tasks. In our study, we explore the use of SKPT for text classification, utilizing a masked language model, since the cloze pre-training task is more appropriate for classification tasks. However, for other generative tasks, it is more suitable to use decoder-based pre-trained models. SKPT can also inject structured knowledge into decoder-based models by modifying the initial manual templates. We will explore SKPT to enhance other models of different architectures.

Author Contributions: Conceptualization, J.L. and L.Y.; methodology, J.L. and L.Y.; software, L.Y.; validation, J.L. and L.Y.; formal analysis, J.L. and L.Y.; investigation, J.L. and L.Y.; resources, J.L. and L.Y.; data curation, L.Y.; writing—original draft preparation, J.L. and L.Y.; writing—review and editing, L.Y.; visualization, L.Y.; supervision, J.L. and L.Y.; project administration, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (grant no. U193607).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: All data included in this study are available upon request by contacting the corresponding author.

Conflicts of Interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.

2. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
3. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2383–2392.
4. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [CrossRef]
5. Liu, T.; Hu, Y.; Gao, J.; Sun, Y.; Yin, B. Zero-shot text classification with semantically extended graph convolutional network. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 8352–8359.
6. Dong, B.; Yao, Y.; Xie, R.; Gao, T.; Han, X.; Liu, Z.; Lin, F.; Lin, L.; Sun, M. Meta-information guided meta-learning for few-shot relation classification. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 1594–1605.
7. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
8. Shin, R.; Lin, C.; Thomson, S.; Chen, C., Jr.; Roy, S.; Platanios, E.A.; Pauls, A.; Klein, D.; Eisner, J.; Van Durme, B. Constrained Language Models Yield Few-Shot Semantic Parsers. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online , 7–11 November 2021; pp. 7699–7715.
9. Shin, T.; Razeghi, Y.; Logan IV, R.L.; Wallace, E.; Singh, S. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 4222–4235.
10. Kolluru, K.; Adlakha, V.; Aggarwal, S.; Mausam; Chakrabarti, S. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 3748–3761.
11. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst.* **2015**, *28* , 649–657.
12. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. Dbpedia—A large-scale, multilingual knowledge base extracted from wikipedia. *Semant. Web* **2015**, *6*, 167–195. [CrossRef]
13. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT understands, too. *arXiv* **2023**, arXiv:2103.10385.
14. Han, X.; Zhao, W.; Ding, N.; Liu, Z.; Sun, M. Ptr: Prompt tuning with rules for text classification. *AI Open* **2022**, *3*, 182–192. [CrossRef]
15. Schick, T.; Schütze, H. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 255–269.
16. Razniewski, S.; Yates, A.; Kassner, N.; Weikum, G. Language models as or for knowledge bases. *arXiv* **2021**, arXiv:2110.04888.
17. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual, 1–6 August 2021; pp. 4582–4597.
18. Lester, B.; Al-Rfou, R.; Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021; pp. 3045–3059.
19. Lee, L.; Johnson, M.; Toutanova, K.; Roark, B.; Frermann, L.; Cohen, S.B.; Lapata, M. *Transactions of the Association for Computational Linguistics*; MIT Press: Cambridge, MA, USA, 2017; Volume 5.
20. Gao, T.; Fisch, A.; Chen, D. Making Pre-trained Language Models Better Few-shot Learners. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual, 1–6 August 2021; pp. 3816–3830.
21. Liu, J.; Liu, A.; Lu, X.; Welleck, S.; West, P.; Le Bras, R.; Choi, Y.; Hajishirzi, H. Generated Knowledge Prompting for Commonsense Reasoning. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 3154–3169.
22. Zhai, J.; Zheng, X.; Wang, C.D.; Li, H.; Tian, Y. Knowledge prompt-tuning for sequential recommendation. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 Octover–3 November 2023; pp. 6451–6461.
23. Hu, S.; Ding, N.; Wang, H.; Liu, Z.; Wang, J.; Li, J.; Wu, W.; Sun, M. Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 2225–2240.
24. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
25. Nickel, M.; Tresp, V.; Kriegel, H.P. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 8 June–2 July 2011.
26. Yang, B.; Yih, W.t.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *arXiv* **2014**, arXiv:1412.6575.

27. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
28. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
29. Sun, Z.; Deng, Z.; Nie, J.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
30. Zhang, S.; Tay, Y.; Yao, L.; Liu, Q. Quaternion Knowledge Graph Embeddings. In Proceedings of the NeurIPS, Vancouver, BA, Canada, 8–14 December 2019; pp. 2731–2741.
31. Socher, R.; Chen, D.; Manning, C.D.; Ng, A. Reasoning with neural tensor networks for knowledge base completion. *Adv. Neural Inf. Process. Syst.* **2013**, *26* , 926–934.
32. Wang, X.; Gao, T.; Zhu, Z.; Zhang, Z.; Liu, Z.; Li, J.; Tang, J. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 176–194. [CrossRef]
33. He, B.; Zhou, D.; Xiao, J.; Jiang, X.; Liu, Q.; Yuan, N.J.; Xu, T. BERT-MK: Integrating Graph Contextualized Knowledge into Pre-trained Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 2281–2290.
34. Peters, M.E.; Neumann, M.; Logan, R.; Schwartz, R.; Joshi, V.; Singh, S.; Smith, N.A. Knowledge Enhanced Contextual Word Representations. 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), Hong Kong, China, 3–7 November 2019; pp. 43–54.
35. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph and text jointly embedding. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–19 October 2014; pp. 1591–1601.
36. Sun, T.; Shao, Y.; Qiu, X.; Guo, Q.; Hu, Y.; Huang, X.J.; Zhang, Z. CoLAKE: Contextualized Language and Knowledge Embedding. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 3660–3670.
37. Xu, W.; Fang, M.; Yang, L.; Jiang, H.; Liang, G.; Zuo, C. Enabling language representation with knowledge graph and structured semantic information. In Proceedings of the 2021 International Conference on Computer Communication and Artificial Intelligence (CCAI), Guangzhou, China, 7–9 May 2021; pp. 91–96.
38. RelatedWords. RelatedWords. 2021. Available online: https://relatedwords.org/ (accessed on 9 April 2024).
39. Miller, G.A. WordNet. In Proceedings of the Workshop on Speech and Natural Language—HLT '91, Harriman, NY, USA, 23–26 February 1992. [CrossRef]
40. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
41. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.