Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification

Shengding Hu[1,3], **Ning Ding**[1,3], **Huadong Wang**[1,3], **Zhiyuan Liu**[1,2,3]*,
Juanzi Li[1,2], **Maosong Sun**[1,2,3]
[1]Department of Computer Science and Technology, [2]Institute for Artificial Intelligence,
[3]State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
{hsd20, dingn18}@mails.tsinghua.edu.cn

Abstract

Tuning pre-trained language models (PLMs) with task-specific prompts has been a promising approach for text classification. Particularly, previous studies suggest that prompt-tuning has remarkable superiority in the low-data scenario over the generic fine-tuning methods with extra classifiers. The core idea of prompt-tuning is to insert text pieces, i.e., template, to the input and transform a classification problem into a masked language modeling problem, where a crucial step is to construct a projection, i.e., verbalizer, between a label space and a label word space. A verbalizer is usually handcrafted or searched by gradient descent, which may lack coverage and bring considerable bias and high variances to the results. In this work, we focus on incorporating external knowledge into the verbalizer, forming a *knowledgeable prompt-tuning* (KPT), to improve and stabilize prompt-tuning. Specifically, we expand the label word space of the verbalizer using external knowledge bases (KBs) and refine the expanded label word space with the PLM itself before predicting with the expanded label word space. Extensive experiments on zero and few-shot text classification tasks demonstrate the effectiveness of knowledgeable prompt-tuning.

1 Introduction

Recent years have witnessed the prominence of Pre-trained Language Models (PLMs) (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Raffel et al., 2020; Xu et al., 2021) due to their superior performance on a wide range of language-related downstream tasks such as text classification (Kowsari et al., 2019), question answering (Rajpurkar et al., 2016), and machine reading comprehension (Nguyen et al., 2016). To fathom the principles of such effectiveness of PLMs, researchers have conducted extensive studies and suggested that PLMs have obtained rich knowledge during pre-training (Petroni et al., 2019; Davison et al., 2019; Roberts et al., 2020). Hence, how to stimulate and exploit such knowledge is receiving increasing attention.

One conventional approach to achieve that is fine-tuning (Devlin et al., 2019), where we add extra classifiers on the top of PLMs and further train the models under classification objectives. Fine-tuning has achieved satisfying results on supervised tasks. However, since the extra classifier requires adequate training instances to tune, it is still challenging to apply fine-tuning in few-shot learning (Brown et al., 2020) and zero-shot learning (Yin et al., 2019) scenarios. Originated from GPT-3 (Brown et al., 2020) and LAMA (Petroni et al., 2019, 2020), a series of studies using prompts (Schick and Schütze, 2020a; Liu et al., 2021) for model tuning bridge the gap between pre-training objective and down-stream tasks, and demonstrate that such discrete or continuous prompts induce better performances for PLMs on few-shot and zero-shot tasks.

A typical way to use prompt-tuning is to wrap the input sentence into a natural language template and let the PLM conduct masked language modeling. For instance, to classify the topic of a sentence x: "What's the relation between speed and acceleration?" into the "SCIENCE" category, we wrap it into a template: "A `[MASK]` question: x". The prediction is made based on the probability that the word "science" is filled in the "`[MASK]`" token. The mapping from *label words* (e.g., "science") to the specific class (e.g., class SCIENCE) is called the *verbalizer* (Schick and Schütze, 2020a). Verbalizer bridges a projection between the vocabulary and the label space and is proven to have a great influence on the performance of classification (Gao et al., 2020).

* Corresponding authors: Z.Liu (liuzy@tsinghua.edu.cn)

Most existing works use human-written verbalizers (Schick and Schütze, 2020a, 2021), in which the designers manually think up a single word to indicate each class. However, the human-written verbalizers usually determine the predictions based on limited information. For instance, in the above mentioned example, the naive verbalizer {science}→ SCIENCE means that only predicting the word "science" for the `[MASK]` token is regarded as correct during inference, regardless of the predictions on other relevant words such as "physics" and "maths", which are also informative. Such handcrafted one-one mapping limits the coverage of label words, thus lacking enough information for prediction and also inducing bias into the verbalizer. Therefore, the handcrafted verbalizers are hard to be optimal in prompt-tuning, where the semantics of label words are crucial for predictions.

Some works try to mitigate the disadvantage of handcrafted verbalizer, and propose to search for the best verbalizer(s) using gradient descent (Liu et al., 2021; Schick et al., 2020) and induce a few words that are similar to the class name in terms of word sense but differ in terms of surface forms. However, such optimization-based expansion is difficult to infer words across granularities (e.g. from "science" to "physics"). If we expand the verbalizer of the above example into {science, physics} → SCIENCE, the probability of predicting the true label will be considerably enhanced. Therefore, to improve the coverage and reduce the bias of the verbalizer we present to incorporate external knowledge into the verbalizers to facilitate prompt-tuning, namely, *knowledgeable prompt-tuning* (KPT). Since our expansion is not based on optimization, it will be more favorable for zero-shot learning.

Specifically, KPT contains three steps: construction, refinement, and utilization. (1) Firstly, in the construction stage, we use external KBs to generate a set of label words for each label (in § 3.2). Note that the expanded label words are not simply synonyms of each other, but covers different granularities and perspectives, thus are more comprehensive and unbiased than the class name. (2) Secondly, we use the PLM itself to denoise the expanded label words. For zero-shot learning, we propose to use a *contextualized prior* to remove those words with low prior probability. Since the words from the KB can have dramatically different prior probabilities, we propose a robust calibration method, namely, *contextualized calibration*, to boost the zero-shot performance (in § 3.3). For few-shot learning, we assign a learnable weight to each label word to denoise the knowledgeable verbalizer. (3) Finally, we apply either a vanilla average loss function or a weighted average loss function for the utilization of expanded verbalizers, which map the scores on a set of label words to the scores of the labels.

We conduct extensive experiments on zero-shot and few-shot text classification tasks. The empirical results show the effectiveness of KPT (in § 4). In addition to the promising improvements than regular prompt-tuning, KPT also reduces the prediction variances in few-shot experiments and yields more stable performances (in § 5). We will make the source code publicly available.

2 Related Work

This work focuses on incorporating knowledge into prompt verbalizers. Thus, three groups of research are related to KPT: prompt-tuning, the verbalizer construction, and knowledge-enhanced PLMs. Since we conduct experiments on text classification tasks, we introduce several works of zero-shot and few-shot text classification in § 4.

Prompt-tuning. Since the emergence of GPT-3 (Brown et al., 2020), prompt-tuning has received considerable attention. GPT-3 (Brown et al., 2020) demonstrates that with prompt-tuning and in-context learning, the large-scale language models can achieve superior performance in the low-data regime. The following works (Schick and Schütze, 2020a,b) argue that small-scale language models (Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019) can also achieve decent performance using prompt-tuning. While most of the researches are conducted on text classification or the tasks in SuperGLUE (Wang et al., 2019), some works extend the impact of prompt-tuning into other tasks, e.g., relation extraction (Han et al., 2021; Chen et al., 2021). In addition to using prompt-tuning for various down-stream tasks, prompt is also used to probe knowledge from the PLMs (Petroni et al., 2019, 2020).

Verbalizer Construction. As introduced in § 1, the verbalizer is an important component in prompt-tuning, and existing studies have shown that verbalizers have a strong influence on the performance of prompt-tuning (Holtzman et al., 2021; Gao et al., 2020). Most works use human-written verbalizers (Schick and Schütze, 2020a), which are highly bi-

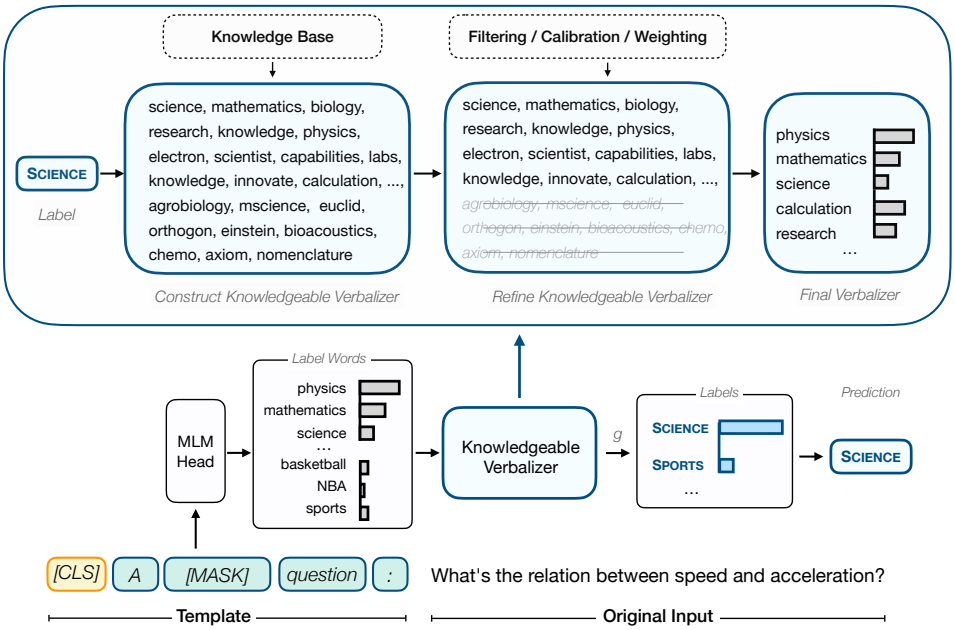

Figure 1: The illustration of KPT , the knowledgeable verbalizer maps the predictions over label words into labels. And the above part is the construction, refinement and utilization processes of KPT .

ased towards personal vocabulary and do not have enough coverage. Some other studies (Gao et al., 2020; Shin et al., 2020; Liu et al., 2021; Schick et al., 2020) design automatic verbalizer searching methods for better verbalizer choices, however, their methods require adequate training set and validation set for optimization. Moreover, the automatically determined verbalizers are usually synonym of the class name, which differs from our intuition of expanding the verbalizer with a set of diverse and comprehensive label words using external KB. Schick et al. (2020); Shin et al. (2020) also try multiple label words for each class. The optimal size of their label words set for each class is generally less than 10. In this work, we propose KPT , which uses external knowledge to boost the performance of prompt-tuning. Compared to the previous strategies, our method can generate and effectively utilize more than 100 related label words across granularities for each class, and can be effectively applied to a zero-shot setting.

Knowledge Enhanced PLMs. Using external knowledge to enhance the performance of PLMs has been extensively studied in recent years, and it is usually applied to the pre-training stage (Zhang et al., 2019b; Liu et al., 2020) and the fine-tuning stage (Yang et al., 2019; Guan et al., 2020). Specifically, in text classification tasks, Chen et al. (2019); Zhang et al. (2019a); Sinoara et al. (2019) also explore utilizing KBs to enhance the input text. Different from these methods, KPT incorporates external knowledge in the prompt-tuning stage and yields remarkable improvements in zero-shot and few-shot text classification tasks.

3 Knowledgeable Prompt-tuning

In this section, we present our methods to incorporate external knowledge into a prompt verbalizer. We first introduce the overall paradigm of prompt-tuning and then elucidate how to construct, refine and utilize the knowledgeable prompt.

3.1 Overview

Let $\mathcal{M}$ be a language model pre-trained on large scale corpora. In text classification task, an input sequence $\mathbf{x} = (x_0, x_1, ..., x_n)$ is classified into a class label $y \in \mathcal{Y}$. Prompt-tuning formalizes the classification task into a masked language modeling problem. Specifically, prompt-tuning wraps the input sequence with a *template*, which is a piece of natural language text. For example, assuming we need to classify the sentence $\mathbf{x}$ ="What's the relation between speed and acceleration?" into label SCIENCE (labeled as 1) or SPORTS (labeled as 2), we wrap it into

$$\mathbf{x}_\text{p} = \texttt{[CLS]}\ \text{A}\ \texttt{[MASK]}\ \text{question}\ :\ \mathbf{x}$$

Then $\mathcal{M}$ gives the probability of each word v in the vocabulary being filled in `[MASK]` token $P_\mathcal{M}(\texttt{[MASK]} = v|\mathbf{x}_\text{p})$. To map the probabilities of words into the probabilities of labels, we define a *verbalizer* as a mapping f from a few words in the vocabulary, which form the *label word* set $\mathcal{V}$,

to the label space $\mathcal{Y}$, i.e., $f: \mathcal{V} \mapsto \mathcal{Y}$. We use $\mathcal{V}_y$ to denote the subset of $\mathcal{V}$ that is mapped into a specific label y, $\cup_{y\in\mathcal{Y}}\mathcal{V}_y = \mathcal{V}$. Then the probability of label y, i.e., $P(y|\mathbf{x}_\text{p})$, is calculated as

$$P(y|\mathbf{x}_\text{p}) = g\left(P_\mathcal{M}(\texttt{[MASK]}=v|\mathbf{x}_\text{p})|v \in \mathcal{V}_y\right), \quad (1)$$

where g is a function transforming the probability of label words into the probability of the label. In the above example, regular prompt-tuning may define $\mathcal{V}_1 = \{$"science"$\}$, $\mathcal{V}_2 = \{$"sports"$\}$ and g as an identity function, then if the probability of "science" is larger than "sports", we classify the instance into SCIENCE.

We propose KPT, which mainly focuses on using external knowledge to improve verbalizers in prompt-tuning. In KPT , we use KBs to generate multiple label words related to each class y, e.g., $\mathcal{V}_1 = \{$"science","physics", ...$\}$. And we propose a contextualized calibration method to eliminate noise in the expanded $\mathcal{V}$. Finally, we explore the vanilla average and weighted average approaches for the utilization of the expanded $\mathcal{V}$. The details are in the following sections.

3.2 Verbalizer Construction

The process of predicting masked words based on the context is not a single-choice procedure, that is, there is no standard correct answer, but abundant words may fit this context. Therefore, the label words mapped by a verbalizer should be equipped by two attributes: *wide coverage* and *little subjective bias*. Such a comprehensive projection is crucial to the imitation of pre-training, i.e., prompt-tuning. Fortunately, external structured knowledge could simultaneously meet both requirements. In this section, we introduce how we use external knowledge for two text classification tasks: topic classification and sentiment classification.

For topic classification, the core issue is to extract label words related to the topic from all aspects and granularities. From this perspective, we choose Related Words [1], a knowledge graph $\mathcal{G}$ aggregated from multiple resources, including word embeddings, ConceptNet (Speer et al., 2017), WordNet (Pedersen et al., 2004), etc., as our external KB. The edges denote "relevance" relations and are annotated with relevance scores which could be used to measure the correlations between label words and topics. We use the name of each topic v as the anchor node to get the neighborhood nodes $N_\mathcal{G}(v)$ whose scores are larger than a threshold η as the related words. Thus, each class is mapped into a set of label words $\mathcal{V}_y = N_\mathcal{G}(v) \cup \{v\}$. For binary sentiment classification, the primary goal is to select as many expressions as possible that tend to be positive or negative. And we use the sentiment dictionary summarized by previous researchers [2,3]. Thus, we get a *knowledgeable verbalizer* mapping multiple label words to a class label, which enhances the handcrafted verbalizer with external knowledge. Several examples of the label words in the KPT are in Table 1.

3.3 Verbalizer Refinement

Although we have constructed a knowledgeable verbalizer that contains comprehensive label words, the collected knowledgeable verbalizer can be very noisy since the vocabulary of the KB is not tailored for the PLM. Thus it is necessary to further refine such verbalizer by retaining high-quality words and removing low-relevance words. In this section, we introduce the refinement of verbalizers in zero-shot and few-shot settings.

Zero-shot Refinement. In zero-shot learning, three problems need to be addressed to facilitate the use of knowledgeable verbalizers. First of all, some of the words recommended by the KB are out-of-vocabulary (OOV) for the PLM, however, these words may also provide information for classification and should not be removed completely. To support the prediction of these words, we simply use the average probability of each token in their tokenizations being filled in the masked position as the probability for these words.

The second problem is to handle the rare words. We assume that several words in the KB are rare to the PLM, thus the prediction probabilities on these words tend to be inaccurate. Instead of using a word-frequency dictionary, we propose to use *contextualized prior* of the label words to remove these words. Specifically, given a text classification task, we denote the distribution of the sentences $\mathbf{x}$ in the corpus as $\mathcal{D}$. For each sentence in the distribution, we wrap it into the template and calculate the predicted probability for each label word v in the masked position $P_\mathcal{M}(\texttt{[MASK]}=v|\mathbf{x}_\text{p})$. By taking the expectation of the probability over the entire distribution of sentences, we can get the prior dis-

[1] https://relatedwords.org

[2] https://www.enchantedlearning.com/wordlist/positivewords.shtml

[3] https://www.enchantedlearning.com/wordlist/negativewords.shtml

Dataset	Label	Label Words
AG's News	POLITICS	politics, government, diplomatic, law, aristotle, diplomatical, governance, ...
	SPORTS	sports, athletics, gymnastics, sportsman, competition, cycling, soccer, ...
IMDB	NEGATIVE	abysmal, adverse, alarming, angry, annoy, anxious, apathy, appalling, ...
	POSITIVE	absolutely, accepted, acclaimed, accomplish, accomplishment, ...

Table 1: Examples of the expanded label words. In topic classification (e.g. AG's News), they are expanded by knowledge graphs, and in sentiment classification (e.g. IMDB), they are expanded by sentiment dictionary.

tribution of the label words in the masked position. We can formalize it as

$$P_{\mathcal{D}}(v) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} P_{\mathcal{M}}(\texttt{[MASK]}=v|\mathbf{x}_{\text{p}}). \quad (2)$$

Empirically, we found that using a small-size *unlabeled support set* $\tilde{\mathcal{C}}$ sampled from the training set and with labels removed, will yield a satisfying estimate of the above expectation. Thus, assuming that the input samples $\{\mathbf{x} \in \tilde{C}\}$ have a uniform prior distribution, the contextualized prior is approximated by

$$P_{\mathcal{D}}(v) \approx \frac{1}{|\tilde{\mathcal{C}}|} \sum_{\mathbf{x}\in\tilde{\mathcal{C}}} P_{\mathcal{M}}(\texttt{[MASK]}=v|\mathbf{x}_{\text{p}}). \quad (3)$$

Then we remove the label words whose prior probabilities are less than a threshold.

The third problem is the drastic difference in the prior probabilities of label words. As previous works (Zhao et al., 2021; Holtzman et al., 2021) have shown, some label words are less likely to be predicted than the others, regardless of the label of input sentences, resulting in a biased prediction. In our setting, the label words in the KB tend to have more diverse prior probabilities. Therefore, we use the contextualized prior of label words to calibrate the predicted distribution, namely, *contextualized calibration* (CC):

$$\tilde{P}_{\mathcal{M}}(\texttt{[MASK]}=v|\mathbf{x}_{\text{p}}) = \frac{P_{\mathcal{M}}(\texttt{[MASK]}=v|\mathbf{x}_{\text{p}})}{P_{\mathcal{D}}(v)}. \quad (4)$$

Compared to the contextual calibration (Zhao et al., 2021) and PMI_{DC} (Holtzman et al., 2021), our method utilizes a small unlabeled support set but yields better and stabler results (see § 5.1).

Few-shot Refinement. In few-shot learning, the refinement is easier since we can identify each label word's influence on the prediction. After collecting the label words from the KB, we first remove the label words that are split into multiple tokens, since they tend to be more tricky to handle in the training objective. To mitigate the problem of noisy label words, we assign a learnable weight w_v to each label word v. The weights form a vector $\mathbf{w} \in \mathbb{R}^{|\mathcal{V}|}$, which is initialized to be a zero vector. The weights are normalized within each $\mathcal{V}_y$:

$$\alpha_v = \frac{\exp(w_v)}{\sum_{u\in\mathcal{V}_y}\exp(w_u)}. \quad (5)$$

Intuitively, in the training process, a small weight is expected to be learned for a noisy label word to minimize its influence on the prediction. Note that in few-shot setting, we do not conduct calibration since the probability of a label word can be trained to the desired magnitude, i.e., $\tilde{P}_{\mathcal{M}}(\texttt{[MASK]} = v|\mathbf{x}_{\text{p}}) = P_{\mathcal{M}}(\texttt{[MASK]}=v|\mathbf{x}_{\text{p}})$.

3.4 Verbalizer Utilization

The final problem is how to map the predicted probability on each refined label word to the decision of the class label y, i.e., the objective function g of the knowledgeable verbalizer. Moreover, in few-shot learning, an additional question is how to optimize the knowledgeable verbalizer.

Average. After refinement, we can assume that each label word of a class contributes equally to predicting the label. Therefore, we use the average of the predicted scores on $\mathcal{V}_y$ as the predicted score for label y. The predicted label $\hat{y}$ is

$$\hat{y} = \text{argmax}_{y\in\mathcal{Y}} \Big(\frac{1}{|\mathcal{V}_y|} \sum_{v\in\mathcal{V}_y} \tilde{P}_{\mathcal{M}}(\texttt{[MASK]}=v|\mathbf{x}_{\text{p}})\Big). \quad (6)$$

We use this method in zero-shot learning since there is no parameter to be trained.

Weighted Average. In few-shot text classification, we adopt a weighted average of label words' scores as the prediction score. We use the refinement weights α_i as the weights for averaging. Thus, the predicted label $\hat{y}$ is

$$\hat{y} = \text{argmax}_{y\in\mathcal{Y}} \frac{\exp\big(s(y|\mathbf{x}_{\text{p}})\big)}{\sum_{y'}\exp\big(s(y'|\mathbf{x}_{\text{p}})\big)}, \quad (7)$$

where $s(y|\mathbf{x}_{\text{p}})$ is

$$s(y|\mathbf{x}_{\text{p}}) = \sum_{v\in\mathcal{V}_y} \alpha_v \log P_{\mathcal{M}}(\texttt{[MASK]}=v|\mathbf{x}_{\text{p}}). \quad (8)$$

This objective function is suitable for continuous optimization by applying a cross-entropy loss on the predicted probability.

4 Experiments

We evaluate KPT on four text classification datasets to demonstrate the effectiveness of incorporating external knowledge into prompt-tuning.

4.1 Datasets and Templates

We carry out experiments on two topic classification datasets: AG's News (Zhang et al., 2015) and DBPedia (Lehmann et al., 2015), and two sentiment classification datasets: IMDB (Maas et al., 2011) and Amazon (McAuley and Leskovec, 2013). The statistics of the datasets are shown in Table 2. The detailed information is in Appendix A.

Name	Type	# Class	Test Size
AG's News	Topic	4	7600
DBPedia	Topic	14	70000
Amazon	Sentiment	2	10000
IMDB	Sentiment	2	25000

Table 2: The statistics of each dataset.

Due to the rich expert knowledge contained, the manual templates are proven to be competitive with or better than auto-generated templates (Gao et al., 2020) even though they are simpler to be constructed. Therefore we use manual templates in our experiments. Manual templates are also more applicable than auto-generated templates in the zero-shot setting. To mitigate the influence of different templates, we test KPT under multiple templates for each dataset. Specifically, we use four manual templates for each dataset which are either introduced by (Schick and Schütze, 2020a) or tailored to fit the dataset. We report both the average results of the four templates and the results of the best template. The specific templates we use for each dataset are in Appendix A.

4.2 Experiment Settings

For the PLM, we use RoBERTa$_{\text{large}}$ (Liu et al., 2019) for all experiments. For test metrics, we use Micro-F1 in all experiments. We have different settings for zero-shot and few-shot experiments.

Zero-shot Experiments. The size of the unlabeled support set $|\tilde{C}|$ is 200. For topic classification, threshold for removing rare words are 0.5. For the sentiment classification dataset, we find that our sentiment dictionary is of high quality, thus we do not remove the words based on prior probability. Since the choices of $\tilde{C}$ will influence the test performance, we repeat 5 times of each experiment in KPT and PT+CC using different random seeds.

Few-shot Experiments. We conduct 5, 10 and 20-shot experiments. For a k-shot experiment, we sample k instances of each class from the original training set to form the few-shot training set and sample another k instances per class to form the validation set. We tune the entire model for 5 epochs and choose the checkpoint with the best validation performance to test. Since the different choices of the few-shot training set and validation set affect the test performance heavily, we repeat the experiments on 5 random seeds.

Other hyper-parameters for tuning the Roberta model can be found in Appendix B.

4.3 Baselines

In this subsection, we introduce the baselines we compare with, including the regular prompt-tuning, prompt-tuning combined with contextualized calibration, and fine-tuning. We also include the reported scores of LOTClass and UDA since they are state-of-the-art of unsupervised and semi-supervised text classification. However, they use much more training resource than KPT , which may lead to unfair comparisons.

Prompt-tuning (PT). The regular prompt-tuning method wraps an input sentence into a hand-crafted template. Different from KPT , it uses the class name as the only label word for each class, which is adopted by PET and most existing works. Note that PET uses several other tricks such as self-training, prompt ensemble, etc. We do not use any of these tricks since we want to study the effect of knowledgeable verbalizers alone. These tricks are orthogonal to our contributions and can be combined into ours in future work.

Prompt-tuning + Contextualized Calibration (PT + CC). This approach is the prompt-tuning combined with the proposed contextualized calibration. We use the same unlabeled support set as KPT to calculate the contextualized prior of label words. This baseline is to see how much improvement is made by contextualized calibration instead of knowledgeable verbalizers. In few-shot learning experiments, we do not include this baseline since we find that calibration is less important for few-shot learning.

Fine-tuning (FT). The traditional fine-tuning method inputs the hidden embedding of `[CLS]` token of the PLM into the classification layer to make predictions. Note that fine-tuning can not be applied to the zero-shot setting, since the classifica-

Method	AG's News	DBPedia	Amazon	IMDB
LOTClass†	82.2	86.0	85.3	80.2
PT	75.1 ± 6.2 (79.1)	67.4 ± 3.6 (71.1)	80.5 ± 9.3 (88.2)	86.4 ± 4.2 (92.5)
PT + CC	80.0 ± 0.8 (81.1)	75.1 ± 5.4 (82.3)	91.1 ± 1.8 (93.7)	90.6 ± 3.1 (93.7)
KPT	**83.0 ± 1.7 (85.9)**	**82.5 ± 4.4 (87.2)**	**92.5 ± 1.3 (94.7)**	**91.5 ± 3.0 (94.2)**

Table 3: Results of zero-shot text classification. Average results and the variances of four templates are shown. The results of the best templates are shown in the brackets. Note that for PT+CC and KPT, we repeat each experiment five times using different random seeds. †means they use different training resources than our setting.

tion layer is randomly initialized.

LOTClass. LOTClass (Meng et al., 2020) uses a PLM to extract the topic-related words from the whole unlabeled training corpus. Then it uses a Masked Category Prediction task to train on the unlabeled corpus with pseudo labels.

UDA. UDA (Xie et al., 2019) uses a small labeled corpus and a large unlabeled corpus. To leverage the unlabeled corpus, they use advanced data-augmentation methods, such as back-translation, to encourage the consistency of predictions over augmented data samples.

4.4 Main Results

In this subsection, we introduce the specific results and provide possible insights of KPT .

Zero-shot. From Table 3, we see that KPT consistently outperforms PT and PT+CC baselines, which indicates the effectiveness of our methods. We achieve superior performance to LOTClass either with average performance of all templates or the best-performance template, even though we do not leverage the large unlabeled training set. Specifically, we observe that the performance boost compared to the baselines in topic classification is higher than sentiment classification, which we conjecture that topic classification requires more external knowledge than sentiment classification. While CC offers huge improvement over PT baseline, the incorporation of external knowledge improves over PT+CC up to 7.4 on DBPedia.

Few-shot. From Table 5, we find KPT consistently outperforms baseline method PT, especially in 5-shot and 10-shot experiments. For 20-shot, we hypothesis that the number of labeled instances is enough to optimize the label words' embeddings away from their original word embeddings so that the rich semantics in the knowledgeable verbalizer may bring less assistance. However, KPT still achieves improvement on three datasets. From the table, we can see that FT is highly unstable in low-shot regime, but with enough data, e.g., 280 data points in total for DBPedia, it is superior to the best template of PT. However, KPT still compares favorably to FT under this setting. Another notable feature of KPT is that it achieves significantly low variances compared with the baseline methods. It is probably because the ensemble of different label words provides a more stable training target. Compared with UDA, although we use significantly less training resource, we are superior to them on AG's News and IMDB.

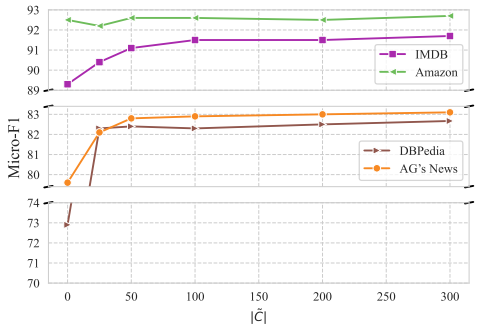

Figure 2: Size of unlabeled support set $\tilde{C}$ w.r.t. test performance. The points at $|\tilde{C}| = 0$ are the performances of PMI_{DC}.

Dataset	Micro-F1	Δ
AG's News	86.8 ± 1.1	-0.4
DBPedia	97.8 ± 0.4	-0.2
IMDB	93.2 ± 0.9	-0.1
Amazon	94.5 ± 1.1	0.1

Table 4: The results of KPT using CC in 10-shot learning. The Δ column shows the differences between the model using CC and the model not using CC.

5 Analysis

In this section, we conduct several ablation studies including the effect of contextualized calibration and the diversity of predicted label words.

5.1 Effect of Contextualized Calibration on Zero-shot Learning

Existing methods propose Domain Conditional PMI (Holtzman et al., 2021) (PMI_{DC}) to calibrate the distribution, which directly measures the prior

shot	Method	AG's News	DBPedia	Amazon	IMDB
	UDA†	86.4	98.6	96.0	88.7
5	FT	37.9 ± 10.0	95.8 ± 1.3	52.1 ± 1.3	51.4 ± 1.4
	PT	83.8 ± 3.1 (85.7)	96.5 ± 0.7 (96.8)	92.8 ± 2.0 (94.6)	92.1 ± 2.4 (94.2)
	KPT	**85.3 ± 0.9 (85.9)**	**97.2 ± 0.6 (97.4)**	**93.3** ± 2.0 (94.6)	**92.5** ± 2.4 **(94.3)**
10	FT	75.9 ± 8.4	93.8 ± 2.2	83.0 ± 7.0	76.2 ± 8.7
	PT	86.3 ± 1.8 (86.5)	97.1 ± 0.8 (97.5)	94.2 ± 1.2 (94.6)	92.8 ± 1.2 **(93.8)**
	KPT	**87.2 ± 0.9 (87.5)**	**98.0 ± 0.3 (98.1)**	**94.4 ± 1.1 (94.8)**	**93.3 ± 0.7** (93.6)
20	FT	85.4 ± 1.8	97.9 ± 0.2	71.4 ± 4.3	78.5 ± 10.1
	PT	87.2 ± 1.8 **(88.4)**	97.5 ± 0.4 (97.6)	94.6 ± 0.9 (94.9)	**93.9 ± 1.0 (94.7)**
	KPT	**87.4 ± 0.9** (88.0)	**98.0 ± 0.2 (98.1)**	**95.0 ± 0.4 (95.3)**	93.8 ± 1.4 (94.5)

Table 5: Results of few-shot text classification. Average Micro-F1 scores and variances using four templates are shown. The Micro-F1 scores of the best templates are shown in the brackets. Note that each experiment is repeated five times using different random seeds. †means they use more training resource than our setting.

probability of label words predicted in the `[MASK]` position given the raw template without filling the template with the instances in the corpus. To compare our method with PMI_{DC} and further assess how many instances are needed to yield a satisfying calibration, we draw the impact of the unlabeled support set's size $|\tilde{C}|$ on the test performance in Figure 2, and draw the performance of PMI_{DC} at $|\tilde{C}| = 0$ for comparison. From Figure 2, we find that $|\tilde{\mathcal{C}}| \sim 100$ is enough to yield a satisfying calibration, and utilizing such a small unlabeled support set produces much better results than PMI_{DC}.

5.2 Is Calibration Important for Few-shot Learning?

Although calibration is crucial for the zero-shot setting, we do not perform calibration for the few-shot setting because we assume that the posterior probability of the label words can be trained to the desired magnitude with only a few training instances. To verify the assumption empirically, we try a 10-shot classification with contextualized calibration. The results and the gap between the methods with and without calibration are reported in Table 4, which indicate that contextualized calibration has little impact in the few-shot scenario.

5.3 Diversity of Top Predicted Words

One advantage of KPT is that it can generate diverse label words across different granularities. To specifically quantify such diversity, we conduct a case study. For the correctly predicted sentences of a class y, we count the frequency of label words $v \in \mathcal{V}_y$ appearing in the top-5 predictions for the `[MASK]` position. Then we report the top-15 frequent label words in Figure 3. Due to space limit, only the results of POLITICS and SPORTS category of AG's News are shown. As shown in Figure 3, a diversity of label words, instead of mainly the original class names, are predicted. And the predicted label words cover various aspects of the corresponding topic. For example, for the topic POLITICS, the predicted "diplomatic", "republic", "parliament" are related to it from different angles.

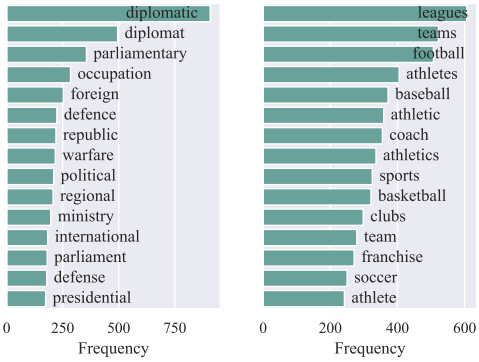

Figure 3: Frequent words appearing in the top-5 predictions. The results for two classes: POLITICS (left) and SPORTS (right) are drawn.

6 Conclusion

In this paper, we propose KPT , which expands the verbalizer in prompt-tuning using the external KB. To better utilize the KB, we propose refinement methods for the knowledgeable verbalizer. The experiments show the potential of KPT in both zero-shot settings and few-shot settings. For future work, there are open questions related to our research for investigation. (1) Sophisticated ways to select the informative label words in the verbalizers. (2) Better approaches for combining KB and prompt-tuning in terms of template construction and verbalizer design. (3) Incorporating external knowledge into prompt-tuning for other tasks such as text generation. We are looking forward to more novel works in this direction.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jindong Chen, Yizhou Hu, Jingping Liu, Yanghua Xiao, and Haiyun Jiang. 2019. Deep short text classification with knowledge powered attention. In *Proceedings of AAAI*, volume 33, pages 6252–6259.

Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Adaprompt: Adaptive prompt-based finetuning for relation extraction. *arXiv preprint arXiv:2104.07650*.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of EMNLP*, Hong Kong, China. Association for Computational Linguistics.

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

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Ari Holtzman, Peter West, Vered Schwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. *arXiv preprint arXiv:2104.08315*.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

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*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of AAAI*, volume 34, pages 2901–2908.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of RecSys*.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *Proceedings of CoCo@ NeurIPS*.

Ted Pedersen, Siddharth Patwardhan, Jason Michelizzi, et al. 2004. Wordnet:: Similarity-measuring the relatedness of concepts. In *Proceedings of AAAI*, volume 4, pages 25–29.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*, pages 2227–2237.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP*, pages 2463–2473.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pages 2383–2392.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of EMNLP*, pages 5418–5426.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of COLING*, pages 5569–5578.

Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of NAACL*, pages 2339–2352.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Roberta A Sinoara, Jose Camacho-Collados, Rafael G Rossi, Roberto Navigli, and Solange O Rezende. 2019. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 163:955–971.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of AAAI*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: a stickier benchmark for general-purpose language understanding systems. In *Proceedings of NeurIPS*, pages 3266–3280.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Han Xu, Zhang Zhengyan, Ding Ning, Gu Yuxian, Liu Xiao, Huo Yuqi, Qiu Jiezhong, Zhang Liang, Han Wentao, Huang Minlie, et al. 2021. Pre-trained models: Past, present and future. *arXiv preprint arXiv:2106.07139*.

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 ACL*, pages 2346–2357.

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

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019a. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of NAACL*, pages 1031–1040.

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

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*.

A Datasets and Templates

In this section, we introduce the information of datasets and the templates we use in detail.

AG's News. AG's News is a news' topic classification dataset. In this dataset, we follow PET (Schick and Schütze, 2020a) to design the templates. However, their best performance pattern $T_1(\mathbf{x})$ = " `[MASK]` news : $\mathbf{x}$" requires the `[MASK]` token to be capitalized, which is not suitable for the label words in KB. And some of their templates are not informative and yield low performances. Therefore, we define four slightly changed templates:

$T_1(\mathbf{x}) =$ A `[MASK]` news : $\mathbf{x}$
$T_2(\mathbf{x}) = \mathbf{x}$ This topic is about `[MASK]`.
$T_3(\mathbf{x}) =$ [Category : `[MASK]`] $\mathbf{x}$
$T_4(\mathbf{x}) =$ [Topic : `[MASK]`] $\mathbf{x}$

DBPedia. In a DBPedia sample, we are given a paragraph $\mathbf{b}$ paired with a title $\mathbf{a}$, in which the title is the subject of paragraph. The task is to determine the topic (or the type) of the subject. Different from other topic classifications, the paragraph can emphasize topics that are different from the title. For example, in a paragraph about an audio company, the main paragraph talks about music, albums, etc., but the correct label is "company" rather than "music". Therefore, we define the following templates:

$T_1(\mathbf{a}, \mathbf{b}) = \mathbf{a}\ \mathbf{b}\ \tilde{\mathbf{a}}$ is a `[MASK]` .
$T_2(\mathbf{a}, \mathbf{b}) = \mathbf{a}\ \mathbf{b}$ In this sentence, $\tilde{\mathbf{a}}$ is a `[MASK]` .
$T_3(\mathbf{a}, \mathbf{b}) = \mathbf{a}\ \mathbf{b}$ The type of $\tilde{\mathbf{a}}$ is `[MASK]`.
$T_4(\mathbf{a}, \mathbf{b}) = \mathbf{a}\ \mathbf{b}$ The category of $\tilde{\mathbf{a}}$ is `[MASK]`.

where $\tilde{\mathbf{a}}$ means removing the last punctuate in the title.

IMDB. IMDB is a sentiment classification dataset about movie reviews. Similar to the template defined in (Schick and Schütze, 2020a) for sentiment classification, we define the following template:

$T_1(\mathbf{x}) =$ It was `[MASK]` .$\mathbf{x}$
$T_2(\mathbf{x}) =$ Just `[MASK]` ! $\mathbf{x}$
$T_3(\mathbf{x}) = \mathbf{x}$ All in all, it was `[MASK]`.
$T_4(\mathbf{x}) = \mathbf{x}$ In summary, the film was `[MASK]`.

Amazon. Amazon is another sentiment classification dataset , we define the following template:

$T_1(\mathbf{x}) =$ It was `[MASK]` .$\mathbf{x}$
$T_2(\mathbf{x}) =$ Just `[MASK]` ! $\mathbf{x}$
$T_3(\mathbf{x}) = \mathbf{x}$ All in all, it was `[MASK]`.
$T_4(\mathbf{x}) = \mathbf{x}$ In summary, it was `[MASK]`".

Since the test set of amazon is unnecessarily large for efficient testing, we randomly sample 10,000 samples from the 400,000 test samples to test, which is proven to have tiny influence on the performance in our pilot experiments.

B Experimental Settings

We list the hyper-parameters in Table 6. Most of the hyper-parameters are the default parameters from Huggingface Transformers[4].

Hyper-parameter	Value
maximum sequence length	512
warmup steps	500
learning rate	3e-5
maximum epochs	5
adam epsilon	1e-8

Table 6: Hyper-parameter settings.

C Detailed Results

To have a close look at the performance of KPT and baselines on each template, we report the performance of each template on zero-shot and 10-shot experiments in Table 7.

[4] https://huggingface.co/transformers/

Shot	Method	Template ID	Agnews	DBPedia	Amazon	IMDB
0	PT+CC	1	81.1 ± 0.2	76.3 ± 0.6	89.2 ± 0.3	88.8 ± 0.1
		2	79.3 ± **0.3**	82.3 ± 0.6	90.8 ± 0.5	86.6 ± **0.1**
		3	80.3 ± 0.3	67.5 ± 0.3	90.7 ± 1.7	93.4 ± 0.5
		4	79.3 ± **0.2**	74.3 ± **0.2**	93.7 ± 0.3	93.7 ± 0.3
		Avg	80.0 ± 0.8	75.1 ± 0.4	91.1 ± 0.7	90.6 ± 0.3
	KPT	1	**85.9** ± **0.1**	**86.5** ± **0.2**	**91.6** ± **0.0**	**91.0** ± **0.1**
		2	**82.3** ± 0.4	**87.2** ± **0.4**	**91.4** ± **0.1**	**87.0** ± 0.2
		3	**82.5** ± **0.3**	**78.8** ± **0.2**	**92.2** ± **0.5**	**93.8** ± **0.3**
		4	**81.4** ± 0.3	**77.8** ± 0.5	**94.7** ± **0.1**	**94.2** ± **0.1**
		Avg	**83.0** ± **0.3**	**82.5** ± **0.3**	**92.5** ± **0.2**	**91.5** ± **0.2**
10	PT	1	86.0 ± 2.7	97.1 ± 0.9	94.4 ± 1.2	93.2 ± 0.6
		2	85.1 ± 2.0	97.5 ± 0.6	93.6 ± 1.7	91.2 ± 0.5
		3	**86.9** ± **1.1**	97.3 ± 0.7	94.6 ± 0.5	93.2 ± 0.8
		4	86.9 ± 1.1	96.7 ± 1.1	94.1 ± 1.0	**93.8** ± **0.6**
		Avg	86.3 ± 1.8	97.1 ± 0.8	94.2 ± 1.2	92.8 ± 1.2
	KPT	1	**87.4** ± **0.6**	**97.8** ± **0.3**	**94.5** ± **1.6**	**93.5** ± **0.4**
		2	**87.5** ± **0.7**	**97.9** ± **0.2**	**93.8** ± **1.3**	**92.7** ± **0.3**
		3	86.6 ± 1.1	**98.0** ± **0.3**	**94.8** ± **0.5**	**93.6** ± **0.4**
		4	**87.1** ± **1.1**	**98.1** ± **0.2**	**94.5** ± **0.6**	93.5 ± 1.0
		Avg	**87.2** ± **0.9**	**98.0** ± **0.3**	**94.4** ± **1.1**	**93.3** ± **0.7**

Table 7: Results of each template in zero-shot and 10-shot text classification.Not that the variances are small within different choices of random seeds for the same template in zero-shot learning. In order not to let the differences across the templates dominate the variances, in the "Avg" row of zero-shot classification, the variance is the average of the variances of different templates, instead of the variance of all experiments.