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

Anonymous ACL submission

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). 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 prompts 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 $\mathbf{x}$: "What's the relation between speed and acceleration?" into the "SCIENCE" category, we wrap it into a template: "A `[MASK]` question: $\mathbf{x}$". 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), which bridges a projection between the vocabulary and the label space and has a great influence on the performance of classification (Gao et al., 2020).

Most existing works use manual verbalizers (Schick and Schütze, 2020a, 2021), in which the designers manually think up a single word to indicate each class. To ease the human effort of designing the class name, some works propose to learn the label words using discrete search (Schick et al.,

2020) or gradient descent (Liu et al., 2021; Hambardzumyan et al., 2021). However, the learned-from-scratch verbalizer, lack of human prior knowledge, is still *considerably inferior* to the manual verbalizers (see Appendix A for pilot experiments), especially in few-shot setting, and even not applicable in zero-shot setting, which leaves the manual verbalizer a decent choice in many cases.

However, manual verbalizers usually determine the predictions based on limited information. For instance, in the above example, the mapping {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 introducing bias into the verbalizer. Therefore, manual verbalizers are hard to be optimal in text classification, where the semantics of label words are crucial for predictions.

The optimization-based expansion, though can be combined with manual verbalizers to yield better performance, only induces a few words or embeddings that are close to the class name in terms of word sense or embedding distance. Thus they are difficult to infer words across granularities (e.g. from "science" to "physics"). If we can expand the verbalizer of the above example into {science, physics} → SCIENCE, the probability of making correct predictions will be considerably enhanced. Therefore, to improve the coverage and reduce the bias of the manual 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 also 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 cover different granularities and perspectives, thus are more comprehensive and unbiased than the class name. (2) Secondly, to cope with the noise in the unsupervised expansion of label words, we propose four refinement methods, namely, frequency refinement, relevance refinement, contextualized calibration, and learnable refinement (in § 3.3), whose effectiveness is studied thoroughly in § 4. (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 that KPT can reduce the error rate of classification by 16%, 18%, 10%, 7% on average in 0, 1, 5, 10 shot experiments, respectively, which shows the effectiveness of KPT. In addition to the performance boost, KPT also reduces the prediction variances consistently in few-shot experiments and yields more stable performances. [1]

2 Related Work

Two groups of research are related to KPT: prompt-tuning, and the verbalizer construction.

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. Prompt-tuning has been applied to a large variety of tasks such as Text Classification (Schick and Schütze, 2020a), Natural Language Understanding (Schick and Schütze, 2020b; Liu et al., 2021) , Relation Extraction (Han et al., 2021; Chen et al., 2021), and Knowledge Probing (Petroni et al., 2019; Liu et al., 2021), etc.

Verbalizer Construction. As introduced in § 1, the verbalizer is an important component in prompt-tuning and has 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 biased 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

[1]The source code will be available upon acceptance.

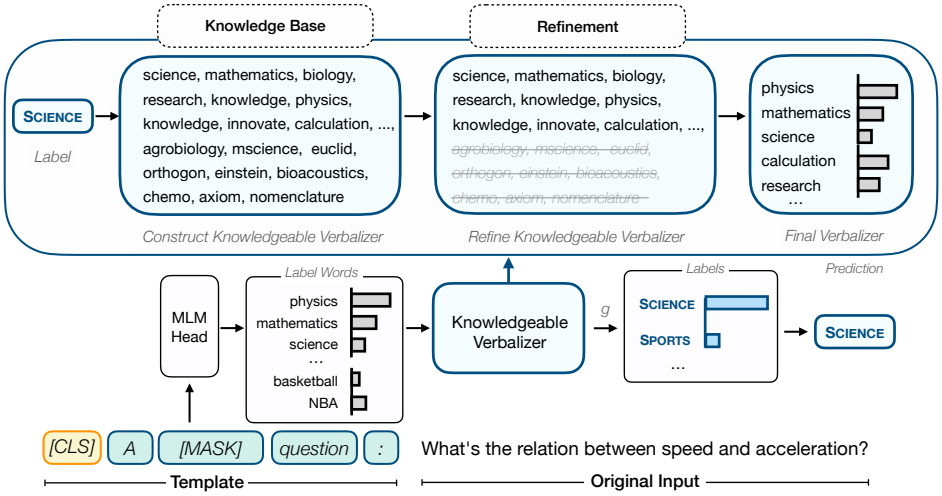

Figure 1: The illustration of KPT , the knowledgeable verbalizer maps the predictions over label words into labels. And the above part is the construction, refinement and utilization processes of KPT .

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, which lacks coverage when used in 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 $\mathcal{V}_1 = \{\text{"science"}\}$, $\mathcal{V}_2 = \{\text{"sports"}\}$ 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., $\mathcal{V}_1 = \{\text{"science"}, \text{"physics"}, ...\}$. And we propose four refinement methods to eliminate the 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, which is

the essence of 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 [2], 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. We presume the the name of each class v_0 is correct and use them as the anchor node to get the neighborhood nodes $N_{\mathcal{G}}(v_0)$ whose scores are larger than a threshold η as the related words [3]. Thus, each class is mapped into a set of label words $\mathcal{V}_y = N_{\mathcal{G}}(v_0) \cup \{v_0\}$. For binary sentiment classification, the primary goal is to extend the binary sentiment to sentiment of more granualities and aspects. We use the sentiment dictionary summarized by previous researchers [4,5]. Several examples of the label words in the KPT are in Table 1.

Dataset	Label	Label Words
AG's News	POLITICS	politics, government, diplomatic, law ...
	SPORTS	sports, athletics, gymnastics, sportsman ...
IMDB	NEGATIVE	abysmal, adverse, alarming, angry, ...
	POSITIVE	absolutely, accepted, acclaimed, ...

Table 1: Examples of the expanded label words.

3.3 Verbalizer Refinement

Although we have constructed a knowledgeable verbalizer that contains comprehensive label words, the collected label words can be very noisy since the vocabulary of the KB is not tailored for the PLM. Thus it is necessary to refine such verbalizer by retaining high-quality words. In this section, we propose four refinement methods addressing different problems of the noisy label words.

Frequency Refinement. The first 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 distribution of the label words in the masked position. We 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. Details can be found in Appendix C.

Relevance Refinement. As our construction of knowledgeable label words is fully unsupervised, some label words may be more relevant to their belonging class than the others. To measure the relevance of a label word to each class, we obtain the prediction probability of the label word on the support set $\tilde{\mathcal{C}}$ as the vector representation $\mathbf{q}^v$ of the label words, i.e., $\mathbf{q}^v$'s i-th element is

$$\mathbf{q}^v_i = P_{\mathcal{M}}([\texttt{MASK}] = v|\mathbf{x}_{i\text{p}}), \mathbf{x}_i \in \tilde{\mathcal{C}}. \quad (4)$$

To estimate the class's representation, we presume that the name of each class v_0, such as "science" for SCIENCE, though lack of coverage, is very relevant to the class. Then we use the vector representation $\mathbf{q}^{v_0}$ of the these names as the class's representation $\mathbf{q}^y$. Therefore the relevance score between a label word v and a class y is calculated as the cosine similarity between the two representation:

$$r(v, y) = \cos(\mathbf{q}^v, \mathbf{q}^y) = \cos(\mathbf{q}^v, \mathbf{q}^{v_0}). \quad (5)$$

Moreover, some label words may contribute positively to multiple classes, resulting in confusion between classes. For example, the potential label

[2] https://relatedwords.org
[3] We take $\eta = 0$ in the experiments
[4] https://www.enchantedlearning.com/wordlist/positivewords.shtml
[5] https://www.enchantedlearning.com/wordlist/negativewords.shtml

word "physiology" of class SCIENCE may also be assigned with a high probability in a sentence of class SPORTS. To mitigate such confusion and filter the less relevant label words, we design a metric that favors the label word with high relevance *merely* to its belonging class and low relevance to other classes:

$$R(v) = r(v, f(v)) \frac{|\mathcal{Y}| - 1}{\sum_{y \in \mathcal{Y}, y \neq f(v)} (r(v, y))}, \quad (6)$$

where $f(v)$ is the corresponding class of v.

Ideally, a good label word should at least has a higher relevance score for its belonging class than the average relevance score for the other classes. Therefore, we remove the label words with $R(v) < 1$. In practice, we have a slight modification to Equation 6, please refer to appendix C for details.

Essentially, this Relevance Refinement adopts the idea of the classical TFIDF (Jones, 1972) algorithm which estimates the relevance of a word to a document. It prefers to use a word that is relevant to a specific document while irrelevant to other documents as the keyword of the document. In KPT, a class is analogous to a document, while a label word is comparable to the word in the document. From this perspective, equation 6 is a variant of TFIDF metric.

Contextualized Calibration. 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, resulting in a severer problem (see Table 2). Therefore, we use the contextualized prior of label words to calibrate the predicted distribution, namely, *contextualized calibration* (CC):

$$\tilde{P}_{\mathcal{M}}(\texttt{[MASK]}{=}v|\mathbf{x}_\text{p}) \propto \frac{P_{\mathcal{M}}(\texttt{[MASK]}{=}v|\mathbf{x}_\text{p})}{P_{\mathcal{D}}(v)}, \quad (7)$$

where $P_{\mathcal{D}}(v)$ is the prior probability of the label word. The final probability is normalized to 1.

Learnable Refinement. In few-shot learning, the refinement can be strengthen by a learning process. Specifically we assign a learnable weight w_v to each label word v (may be already refined by the previous methods). 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 $\mathcal{V}_y$:

$$\alpha_v = \frac{\exp(w_v)}{\sum_{u \in \mathcal{V}_y} \exp(w_u)}. \quad (8)$$

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, calibration may not be necessary because the probability of a label word can be trained to the desired magnitude, i.e., $\tilde{P}_{\mathcal{M}}(\texttt{[MASK]}{=}v|\mathbf{x}_\text{p}) = P_{\mathcal{M}}(\texttt{[MASK]}{=}v|\mathbf{x}_\text{p})$.

In addition to these refinement methods, since many label words are out-of-vocabulary for the PLM and are split into multiple tokens by the tokenizer. For these words, we simply use the average prediction score of each token as the prediction score for the word. The influence of this simple approach is studied in Appendix E.3.

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.

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

$$\hat{y} = \text{argmax}_{y \in \mathcal{Y}} \Big(\frac{1}{|\mathcal{V}_y|} \sum_{v \in \mathcal{V}_y} \tilde{P}_{\mathcal{M}}(\texttt{[MASK]}{=}v|\mathbf{x}_\text{p}) \Big). \quad (9)$$

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

Weighted Average. In few-shot setting, supported by the Learnable Refinement, we adopt a weighted average of label words' scores as the prediction score. The refinement weights is used α_i as the weights for averaging. Thus, the predicted $\hat{y}$ is

$$\hat{y} = \text{argmax}_{y \in \mathcal{Y}} \frac{\exp\left(s(y|\mathbf{x}_\text{p})\right)}{\sum_{y'} \exp\left(s(y'|\mathbf{x}_\text{p})\right)}, \quad (10)$$

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

$$s(y|\mathbf{x}_\text{p}) = \sum_{v \in \mathcal{V}_y} \alpha_v \log P_{\mathcal{M}}(\texttt{[MASK]}{=}v|\mathbf{x}_\text{p}). \quad (11)$$

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

3.5 Theoretical Illustration of KPT

We provide a theoretical illustration of the KPT framwork in Appendix B.

4 Experiments

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

Method	AG's News	DBPedia	Yahoo	Amazon	IMDB
PT	75.1 ± 6.2 (79.0)	66.6 ± 2.3 (68.4)	45.4 ± 7.0 (52.0)	80.2 ± 8.8 (87.8)	86.4 ± 4.0 (92.0)
PT+CC	79.9 ± 0.7 (81.0)	73.9 ± 4.9 (82.6)	58.0 ± 1.4 (58.8)	91.4 ± 1.6 (93.5)	91.6 ± 3.0 (93.7)
KPT	**84.8 ± 1.2** (**86.7**)	**82.2 ± 5.4** (**87.4**)	**61.6 ± 2.2** (**63.8**)	**92.8 ± 1.2** (**94.6**)	**91.6 ± 2.7** (94.0)
-FR	82.7 ± 1.5 (85.0)	81.8 ± 4.6 (86.2)	60.9 ± 1.5 (62.7)	**92.8 ± 1.2** (**94.6**)	91.6 ± 2.8 (**94.1**)
-RR	81.4 ± 1.5 (83.7)	81.4 ± 4.5 (85.8)	60.1 ± 1.0 (61.4)	**92.8 ± 1.2** (**94.6**)	91.6 ± 2.8 (**94.1**)
-CC	55.5 ± 2.8 (58.3)	64.5 ± 6.8 (73.0)	42.4 ± 5.0 (46.8)	86.2 ± 5.7 (92.5)	90.3 ± 2.8 (**94.1**)

Table 2: Results of zero-shot text classification. The results of the best templates are shown in the brackets. Indentation means that the experimental configuration is a modification based on the up-level indentation.

4.1 Datasets and Templates

We carry out experiments on three topic classification datasets: AG's News (Zhang et al., 2015), DBPedia (Lehmann et al., 2015), and Yahoo (Zhang 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 8. The detailed information and the statistics of each dataset is in Appendix F.

We test all prompt-based methods using four manual templates and report both the average results (with standard error) of the four templates and the results of the best template (shown in (brackets)). The reasons for using manual templates and the specific templates for each dataset are in Appendix F.

4.2 Experiment Settings

Our experiments are based on OpenPrompt (Ding et al., 2021), which is an open-source toolkit to conduct prompt learning. 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. For all zero-shot experiments, we repeat the experiments 3 times using different random seeds if randomness is introduced in the experiments, and for all few-shot experiments, we repeat 5 times. Note that considering the four templates and five/three random seeds, each reported score of prompt-based methods is *the average of 20/12 experiments*, which greatly reduces the randomness of the evaluation results. For the refinement based on the support set $\tilde{\mathcal{C}}$, the size of the unlabeled support set $|\tilde{C}|$ is 200. For few-shot learning, we conduct 1, 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. Other hyper-parameters can be found in Appendix G.

4.3 Baselines

In this subsection, we introduce the baselines we compare with. To better understand our proposed methods, we also compare within the performance of KPT using different configuration.

Fine-tuning (FT). 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 classification layer is randomly initialized.

Prompt-tuning (PT). The regular prompt-tuning method uses the class name as the only label word for each class, which is used in PET (Schick and Schütze, 2020a) and most existing works. For a fair comparison, we do not use the tricks in PET, such as self-training and prompt ensemble, which are orthogonal to our contributions.

Automatic Verbalizer (AUTO). The automatic verbalizer is proposed by PETAL (Schick et al., 2020), which uses *labeled* data to select the most informative label words *inside* a PLM's vocabulary. It is targeted at the situation when no manually defined class names are available. It's not obvious how to combine it with the manually defined class name to boost the performance, and how it can be applied in a zero-shot setting. Therefore we only compare it in the few-shot setting with no class name information given.

Soft Verbalizer (SOFT). The soft verbalizer is proposed by WARP (Hambardzumyan et al., 2021). They use a continuous vector for each class and use the dot product between the masked language model output and the class vector to produce the probability for each class. In our experiments, its class vectors are initialized with the class names' word embedding, since it is more effective with manual class names as the initial values (see Appendix A). As an optimization-based method, Soft Verbalizer is not applicable in the zero-shot setting.

Shot	Method	AG's News	DBPedia	Yahoo	Amazon	IMDB
1	FT	19.8 ± 10.4	8.6 ± 4.5	11.1 ± 4.0	49.9 ± 0.2	50.0 ± 0.0
	PT	80.0 ± 6.0 (84.4)	92.2 ± 2.5 (94.3)	54.2 ± 3.1 (55.7)	91.9 ± 2.7 (93.2)	91.2 ± 3.7 (93.7)
	AUTO	52.8 ± 9.8 (57.6)	63.0 ± 8.9 (68.3)	23.3 ± 4.5 (25.0)	66.6 ± 12.5 (72.7)	75.5 ± 15.5 (83.1)
	SOFT	80.0 ± 5.6 (82.4)	92.3 ± 2.3 (93.3)	54.3 ± 2.7 (55.9)	90.9 ± 5.8 (93.6)	89.4 ± 8.9 (93.1)
	KPT	**83.7 ± 3.5** (**84.6**)	**93.7 ± 1.8** (**95.3**)	**63.2 ± 2.5** (**64.1**)	93.2 ± 1.3 (**93.9**)	92.2 ± 3.0 (**93.6**)
	- LR	83.5 ± 3.8 (84.3)	93.0 ± 1.8 (94.5)	62.2 ± 2.9 (63.6)	**93.3 ± 1.3** (**93.9**)	92.2 ± 2.8 (**93.6**)
	- RR	82.2 ± 3.2 (82.6)	92.9 ± 1.8 (94.1)	61.3 ± 4.2 (62.5)	93.1 ± 1.5 (93.7)	**92.6 ± 1.7** (**93.6**)
	- RR - LR	81.8 ± 3.3 (82.5)	91.3 ± 1.7 (92.6)	60.7 ± 4.2 (61.4)	93.2 ± 1.5 (93.9)	92.6 ± 1.5 (93.5)
5	FT	37.9 ± 10.0	95.8 ± 1.3	25.3 ± 14.2	52.1 ± 1.3	51.4 ± 1.4
	PT	82.7 ± 2.7 (84.0)	97.0 ± 0.6 (**97.3**)	62.4 ± 1.7 (63.9)	92.2 ± 3.3 (93.5)	91.9 ± 3.1 (92.7)
	AUTO	72.2 ± 10.1 (75.6)	88.8 ± 3.9 (91.5)	49.6 ± 4.3 (51.2)	87.5 ± 7.4 (90.8)	86.8 ± 10.1 (92.1)
	SOFT	82.8 ± 2.7 (84.3)	97.0 ± 0.6 (97.2)	61.8 ± 1.8 (63.1)	93.2 ± 1.6 (**94.2**)	91.6 ± 3.4 (**93.9**)
	KPT	85.0 ± 1.2 (**85.9**)	97.1 ± 0.4 (97.3)	**67.2 ± 0.8** (**67.8**)	93.4 ± 1.9 (94.1)	92.7 ± 1.5 (92.9)
	- LR	**85.1 ± 1.0** (85.8)	97.1 ± 0.4 (97.2)	67.0 ± 1.1 (67.5)	93.4 ± 1.9 (94.1)	92.8 ± 1.5 (93.0)
	- RR	84.3 ± 1.8 (84.9)	**97.2 ± 0.4** (**97.3**)	**67.2 ± 0.8** (67.7)	**93.6 ± 1.4** (94.1)	**93.0 ± 2.0** (93.8)
	- RR - LR	84.2 ± 1.7 (84.5)	97.1 ± 0.4 (97.3)	66.6 ± 1.4 (67.5)	93.4 ± 2.0 (94.1)	93.0 ± 2.1 (93.8)
10	FT	75.9 ± 8.4	93.8 ± 2.2	43.8 ± 17.9	83.0 ± 7.0	76.2 ± 8.7
	PT	84.9 ± 2.4 (86.1)	97.6 ± 0.4 (97.8)	64.3 ± 2.2 (64.8)	93.9 ± 1.3 (94.6)	**93.0 ± 1.7** (**94.0**)
	AUTO	81.4 ± 3.8 (84.1)	91.5 ± 3.4 (95.1)	58.7 ± 3.1 (60.9)	93.7 ± 1.2 (94.5)	91.1 ± 5.1 (93.3)
	SOFT	85.0 ± 2.8 (86.7)	97.6 ± 0.4 (97.8)	64.5 ± 2.2 (65.0)	93.9 ± 1.7 (93.9)	91.8 ± 2.6 (93.0)
	KPT	**86.3 ± 1.6** (87.0)	**98.0 ± 0.2** (**98.1**)	**68.0 ± 0.6** (**68.2**)	93.8 ± 1.2 (94.1)	92.9 ± 1.8 (93.3)
	- LR	85.9 ± 1.9 (**87.1**)	98.0 ± 0.2 (**98.1**)	67.9 ± 0.7 (**68.2**)	93.9 ± 1.1 (94.1)	**93.0 ± 1.7** (93.2)
	- RR	85.6 ± 1.4 (86.2)	97.9 ± 0.2 (98.0)	67.5 ± 1.1 (68.1)	**94.0 ± 1.0** (**94.7**)	92.7 ± 2.1 (93.0)
	- RR - LR	85.1 ± 1.4 (86.0)	97.8 ± 0.2 (97.8)	66.8 ± 1.1 (67.6)	94.1 ± 0.9 (94.8)	93.0 ± 2.0 (93.4)

Table 3: Results of 1/5/10-shot text classification. Indentation means that the experimental configuration is a modification based on the up-level indentation. For results of 20-shot experiments, please see Appendix D.

PT+CC. For zero-shot setting, we further introduce PT combined with our proposed contextualized calibration [6] as a baseline to see how much improvement is made by contextualized calibration instead of knowledgeable verbalizers.

For **KPT** , we experiment with different variants to better understand the proposed methods such as refinement. **-FR**, **-RR**, **-CC** and **-LR** is the variant that does not conduct Frequency Refinement, Relevance Refinement, Contextualized Calibration, and Learnable Refinement, respectively. In few-shot experiments, we presume that the supervised training data can train the output probability of each label word to the desired magnitude, thus we don't use CC and FR in the KPT . This decision is justified in Appendix E.2.

4.4 Main Results

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

Zero-shot. From Table 2, we see that all the variants of KPT , except for KPT-CC, consistently outperforms PT and PT+CC baselines, which indicates the effectiveness of our methods. Comparison between PT and PT+CC proves that Contextualized Calibration is very effective in the zero-shot setting. The results of KPT-FR-RR-CC, which is the variant without any refinement, reveal the label noise is severe in the automatically constructed knowledgeable label words. The gap between KPT-FR-RR and KPT-FR-RR-CC is larger than the gap between PT+CC and PT, demonstrating the drastic difference in the prior probabilities of the knowledgeable label words as we hypothesized in § 3.3. Comparison between KPT, KPT-FR, KPT-FR-RR proves the effectiveness of the refinement methods.

For the analysis regarding each type of classification task, 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 (on average +13%) over PT baseline, the incorporation of external knowledge further improves over PT+CC up to 11% on DBPedia, and 6% on AG's News and Yahoo. We also observe that the improvement brought by the refinement methods is more noticeable for topic classification tasks. By looking at the fraction of label words maintained after the refinement process (See appendix E.4), we conjecture that the sentiment dictionary that we used in sentiment clas-

[6] The same support sets are used as KPT

sification tasks contains little noise. Moreover, the improvement brought by the refinement process justifies the resilience of our methods to recover from noisy label words.

Few-shot. From Table 3, we first find out that prompt-based methods win over fine-tuning by a dramatic margin under nearly all situations. The gap enlarges as the shot becomes fewer. Comparing the baseline methods, the Soft Verbalizer (SOFT) generally wins over the Manual Verbalizer(PT) by a slight margin. However, automatic verbalizer (AUTO), although free of manual effort, lags behind the other verbalizers especially in a low-shot setting. The reason is obvious since the selection of label words among the vocabulary becomes inaccurate when labeled data is limited.

When comparing KPT with the baseline methods, we find KPT or its variants consistently outperform all baseline methods. On average, 17.8% , 10.3%, and 7.4% error rate reduction from the best baseline methods are achieved on 1, 5, and 10 shot experiments, respectively. Comparing within the variants of KPT , we find that RR and LR are generally effective across shots on topic classification dataset, while in sentiment classification dataset, KPT works well without the refinements, which is consistent with our previous assumptions that the sentiment dictionary has little noise. Note that the KPT-RR variant does not utilize any unlabeled support set $\tilde{\mathcal{C}}$ since we do not conduct CC and FR by default in few-shot learning. This variant is still superior to the baseline methods in most cases. In terms of variance, we can see that KPT enjoys smaller variances than baseline methods in most cases, demonstrating that the better coverage of label words stabilizes the training.

5 Analysis

Ablation studies about our refinement methods have been shown in the previous section. In this section and Appendix E, we conduct more in-depth analyses on the proposed methods.

5.1 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 2. Due to space limit, only the results of SPORTS and BUSINESS category of AG's News are shown. As shown in Figure 2, 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 SPORTS, the predicted "leagues", "football", "coach" are related to it from different angles.

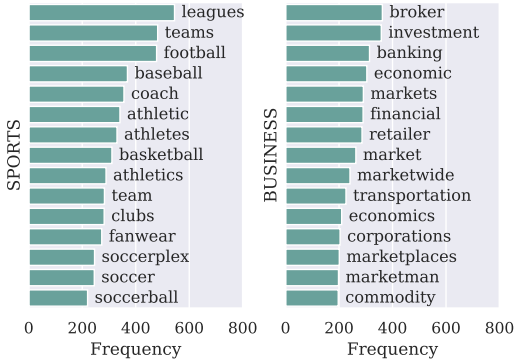

Figure 2: Frequent words appearing in the top-5 predictions. The results for two classes: SPORTS (left) and BUSINESS (right) are drawn.

5.2 Other Analyses

In addition to the visualization, we study the influence of the support set's size on zero-shot text classification in Appendix E.1. Then we justify that few-shot learning via labeled data eases the need for calibration and frequency-based refinement in Appendix E.2. We also demonstrate that our approach to handling the out-of-vocabulary (OOV) words is reasonable in Appendix E.3. Moreover, we take a closer look at the refinement process by analyzing the fraction of label words maintained during refinement in Appendix E.4. Finally, we discuss the potential use of the proposed methods when knowledge bases resources are not readily available in Appendix E.5.

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) Better approaches for combining KB and prompt-tuning in terms of template construction and verbalizer design. (2) Incorporating external knowledge into prompt-tuning for other tasks such as text generation.

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*.

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*.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

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

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of ACL*.

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*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Proceedings of EMNLP*.

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.

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.

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*.

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

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*.

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.

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

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 Pilot Experiments

As pointed out by (Gao et al., 2020), manually defined verbalizer is competitive or even better than automatically searched/optimized verbalizers, which strengthens our motivation to improve over manual verbalizers by injecting more external human knowledge. To further illustrate the advantage of manual verbalizer, we conduct pilot experiments in soft verbalizer. Soft Verbalizer (Hambardzumyan et al., 2021) can be initialized with the predefined class name as the label words, which is adopted by us as a baseline in Table 3. It can also be randomly initialized without the manually defined class names. We test the performance of Soft Verbalizer with and without the manually defined class name in 5 and 10 shot experiments. From Table 4, we can see that the gaps between the variants are generally large. Therefore further improving the verbalizer with manually defined class name is a promising direction than the learned-from-scratch verbalizer without any human prior.

B A Theoretical Illustration of KPT

In this section we provide a theoretical analysis of the whole framework used by KPT . In prompt tuning, given a text $\mathbf{x}$, we wrap it into a template to form a wrapped sentence $\mathbf{x}_\text{p}$. we then predict the probability of the label word v using a PLM:

$$p(\texttt{[M]}{=}v|\mathbf{x}) = P_\mathcal{M}(\texttt{[M]}{=}v|\mathbf{x}_\text{p}), \tag{12}$$

where `[M]` is short for `[Mask]`, denoting the label word's prediction at the masked position of the template.

Then, if multiple label words are used to contribute to a single label, the predicted probability of the label is defined by marginalizing the probability of predicting all the label words, i.e.,

$$p(\text{Y} = y|\mathbf{x}) = \sum_{v \in V_\mathcal{Y}} p(\text{Y} = y, \texttt{[MASK]}{=}v|\mathbf{x}). \tag{13}$$

Since the prediction of Y is independent of $\mathbf{x}$ given v, we can write Equation 13 into

$$\begin{aligned} &\sum_{v \in V_\mathcal{Y}} p(\text{Y} = y|\texttt{[M]}{=}v) p(\texttt{[M]}{=}v|\mathbf{x}) \\ =& \sum_{v \in V_\mathcal{Y}} p(\text{Y} = y|\texttt{[M]}{=}v) P_\mathcal{M}(\texttt{[M]}{=}v|\mathbf{x}_\text{p}). \end{aligned} \tag{14}$$

Using the Bayes Theorem and assuming a balanced classification problem, Equation 15 can be transformed into

$$\begin{aligned} &\sum_{v \in V_\mathcal{Y}} \frac{p(\texttt{[M]}{=}v|\text{Y} = y)p(\text{Y} = y)}{p(\texttt{[M]}{=}v)} P_\mathcal{M}(\texttt{[M]}{=}v|\mathbf{x}_\text{p}) \\ \propto& \sum_{v \in V_\mathcal{Y}} \frac{p(\texttt{[M]}{=}v|\text{Y} = y)}{p(\texttt{[M]}{=}v)} P_\mathcal{M}(\texttt{[M]}{=}v|\mathbf{x}_\text{p}). \end{aligned} \tag{15}$$

Now, the prediction probability of the label is composed of three parts.

(1) The first part $p(v|\text{Y} = y)$ is the probability of predicting the specific label word v given the class label y. Intuitively, if a label word is relevant to label y, this term will be assigned a high probability. In KPT , the **Relevance Refinement** estimate this probability using a quantized objective, i.e., if a relevance score exceeds the threshold 1, it will be maintained, otherwise, it will be filtered. On the other hand, **Learnable Refinement** estimates this probability using continuous weights.

(2) The second part is $p(\texttt{[M]}{=}v)$ in the denominator. This term is actually the prior probability of label words v, which is estimated by our **Contextualized Calibration**. Previous works also try to approach this term using a context-free manner (Holtzman et al., 2021; Zhao et al., 2021).

(3) The last term $P_\mathcal{M}(\texttt{[MASK]}{=}v|\mathbf{x}_\text{p})$ is the probability of the label words v predicted by the PLM, which is the only component in most works such as Manual Verbalizers (Schick and Schütze, 2020a), yielding a sub-optimial solution compared to KPT .

Verbalizers with multiple label words for a class label can all be formalized into this framework once it uses Equation 13 as their backbone hypothesis. However, to the best of our knowledge, KPT is the first to combine all of the three components to form a powerful verbalizer.

C Practical Issues of Refinement

In this section, we detail the refinement process by making some practical modifications to the methods in § 3.3.

Frequency Refinement. For Frequency Refinement, since the absolute value distribution of the contextualized prior probability may be different for each task, determining a specific threshold of the contextualized prior probability may be tricky and elusive. We use a ranking-based threshold, i.e., we filter the label words that appear in the lower half of the contextualized prior probability.

Relevance Refinement. For Relevance Refinement, we observe that in the classification task with

Shot	Method	AG's News	DBPedia	Yahoo	Amazon	IMDB
5	SOFT	82.8 ± 2.7 (84.3)	97.0 ± 0.6 (97.2)	61.8 ± 1.8 (63.1)	93.2 ± 1.6 (94.2)	91.6 ± 3.4 (93.9)
	SOFT w.o. M	63.4 ± 11.3 (64.7)	82.1 ± 5.9 (86.1)	24.5 ± 6.2 (27.2)	79.2 ± 10.5 (85.5)	83.6 ± 11.5 (93.4)
10	SOFT	85.0 ± 2.8 (86.7)	97.6 ± 0.4 (97.8)	64.5 ± 2.2 (65.0)	93.9 ± 1.7 (93.9)	91.8 ± 2.6 (93.0)
	SOFT w.o. M	77.4 ± 4.8 (79.1)	94.9 ± 2.5 (95.9)	42.6 ± 8.3 (48.1)	92.9 ± 2.0 (94.0)	88.7 ± 9.7 (93.8)

Table 4: Pilot experiment on soft verbalizer justifies the need of human (expert) knowledge into the verbalizer. SOFT is the soft verbalizer with class name and SOFT w.o. M is the variant without the manual verbalizer.

Shot	Method	AG's News	DBPedia	Yahoo	Amazon	IMDB
20	FT	85.4 ± 1.8	97.9 ± 0.2	54.2 ± 18.1	71.4 ± 4.3	78.5 ± 10.1
	PT	86.5 ± 1.6 (87.0)	97.9 ± 0.3 (98.1)	67.2 ± 1.1 (67.5)	93.5 ± 1.0 (94.4)	93.0 ± 1.1 (93.6)
	AUTO	85.7 ± 1.4 (86.1)	92.2 ± 2.7 (94.9)	65.0 ± 1.8 (66.9)	**93.9 ± 1.1** (94.1)	92.8 ± 2.0 (**94.0**)
	SOFT	86.4 ± 1.7 (87.1)	98.0 ± 0.3 (98.1)	67.4 ± 0.7 (67.5)	93.8 ± 1.6 (94.2)	**93.5 ± 0.9** (**94.0**)
	KPT	87.2 ± 0.8 (87.5)	**98.1 ± 0.3** (**98.2**)	68.9 ± 0.8 (69.3)	93.7 ± 1.6 (94.4)	93.1 ± 1.1 (93.5)
	- LR	**87.7 ± 0.6** (**87.8**)	**98.1 ± 0.3** (**98.2**)	68.8 ± 0.9 (**69.8**)	93.4 ± 2.3 (94.3)	93.4 ± 0.9 (93.6)
	- RR	87.3 ± 0.8 (87.5)	**98.1 ± 0.3** (**98.2**)	68.8 ± 0.9 (68.9)	93.6 ± 1.3 (94.2)	93.1 ± 0.8 (93.6)
	- RR - LR	87.1 ± 0.9 (87.4)	**98.1 ± 0.3** (**98.2**)	**69.0 ± 0.7** (69.3)	93.7 ± 0.9 (**94.5**)	93.1 ± 0.8 (93.7)

Table 5: Results of 20-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.Indentation means that the experimental configuration is a modification based on the up-level indentation.

only a few classes, it's better to provide a stricter criterion to ensure that the relevance scores of a label word to *any other* class is lower than the score to the belonging class, i.e., *maximum* in the term of IDF-score is preferred. To keep a unified criterion, we use a norm-based IDF-score.

$$R^d(v) = r(v, f(v)) \left(\frac{|\mathcal{Y}| - 1}{\sum_{y \in \mathcal{Y}, y \neq f(v)} (r(v, y)^d)} \right)^{1/d}, \quad (16)$$

where

$$d = \frac{C}{|\mathcal{Y}| - 2 + \epsilon} + 1, C > 0. \quad (17)$$

This criterion will approximate the maximum value in $\{r(v, y) | y \in |Y|, y \neq f(v)\}$ in classification with only a few labels, and revert to the mean score in Equation 6 when conducting classification with many labels. We take $C = 10$ (without trial and error) in the experiments. And $0 < \epsilon \ll 1$ is a small number to prevent numerical error.

D Results of 20 Shot Setting

Due to the space limitation, we report the performance of 20-shot classification in this section. As we can see in Table 5, the gap between different methods narrows as the training data becomes sufficient. However, KPT and its variants still win by a consistent margin over the baseline methods. Surprisingly, with more training data, Learnable Refinement does not become more powerful as we may hypothesize. We conjecture that it is because all label words, even with some noise, *can* serve as training objectives for prompt tuning. This perspective is similar to Gao et al. (2020) that using "bad" as a label word for the class "positive" can still do classification even though the performance decreases.

E Further Analyses and Ablation Studies

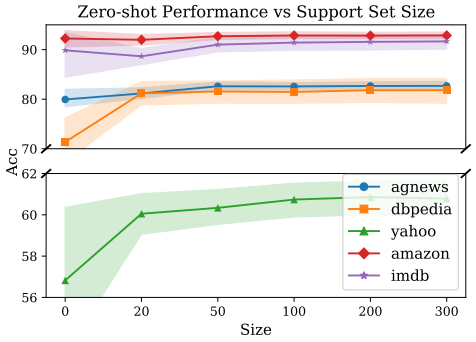

Figure 3: Support set size w.r.t. zero-shot performance. The points at size=0 are the performances of PMI_{DC}.

E.1 Calibration and Contextualized Calibration.

In fine-tuning, calibration has been studied under the topic of prediction confidence and out-of-distribution detection (Kong et al., 2020). Recently, it got renascent attention in prompt learning (Zhao et al., 2021; Holtzman et al., 2021). In prompt learning, the PLM has a natural tendency to predict one word over another word regardless of the real sentence input. For example, GPT-3 prefers to

predict "positive" over "negative" given "N/A" as the input sentence (Zhao et al., 2021). Therefore the calibration is crucial (see Table 2) when no posterior optimization is conducted, i.e., in zero-shot learning. Existing methods such as PMI_{DC} propose only using the empty template without filling the template with the instances in the corpus, for example, "A `[Mask]` question :", to produce the calibration logits. Our proposed Contextualized Calibration utilizes a limited amount of unlabeled support data to yield significantly better results. However, since we target the data-scarce scenario, we study in detail the amount of unlabeled data necessary to produce a satisfying calibration result. In Figure 3, we draw the performance of KPT - RR with different support set sizes $|\tilde{\mathcal{C}}|$. We also draw the performance of PMI_{DC} on the $|\tilde{\mathcal{C}}| = 0$ for comparison.

From Figure 3, we find that $|\tilde{\mathcal{C}}| \sim 50$ is enough to yield a satisfying calibration. Contextualized calibrate is more effective in classification with many classes, while calibrate without the context is effective in classification with few classes.

In addition, we *must point out* that if we have a set of sentences to classify in real-world scenarios, we can use these sentences themselves as the support set to conduct more accurate Contextualized Calibration.

E.2 Supervised Data Ease the Need for Calibration.

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. We also do not perform Frequency Refinement for few-shot learning due to the same assumption. To verify the assumption empirically, we add both Contextualized Calibration and Frequency Refinement to KPT and test the performance under different settings. The results are shown in Table 6. The performance comparison to KPT without CC and FR in Table 3 and Table 5 are shown using up arrows and down arrows. We can see that except in Yahoo, the improvement is not consistent for even negative, which supports our assumption that the need for calibration is greatly eased with the supervised input data.

E.3 How to Handle the OOV Label Words?

Since the knowledgeable verbalizer is expanded using external resources which may not be tailored for the vocabulary of PLM. Thus, many label words are out-of-vocabulary (OOV) and are split into multiple tokens by the tokenizer. For these words, as mentioned in § 3.3, we average the prediction probability of each token in the *single* `[MASK]` position, which may not be very reasonable at the first glance. Therefore, we conduct an ablation study that whether forcing the label words to be a single token in the vocabulary of the PLM leads to better performance. The results under different shots are shown in Table 7. Surprisingly, making the single-token restriction does not yield stable improvement, instead, in many cases, the performance degrades by minor margins. Therefore we conclude that our method to handle OOV label words that are split by the tokenizer into multiple tokens is simple yet reasonable. More importantly, the label words expanded by the knowledge bases but not in the PLM's vocabulary *can* serves as good label words in prompt tuning as well.

E.4 Visualization of the Refinement Process.

In this section, we report the number of label words that remained after Frequency Refinement and Relevance Refinement process. As we can see, these refinement methods remove a large fraction of label words while retaining the ones that are most informative. However, even the fewest number of remaining label words exceeds 100, which is far more than the number of label words in the previous works (Schick et al., 2020). The broad coverage of label words contributes to the success of KPT .

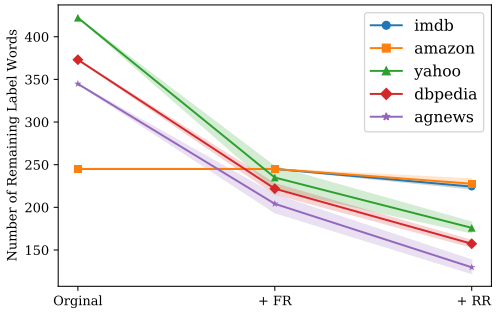

Figure 4: The number of remaining label words after Frequency Refinement and Relevance Refinement.

Shot	Method	AG's News	DBPedia	Yahoo	Amazon	IMDB
1	KPT + CC + FR	83.4 ↓ ± 4.0 (84.6)	94.0 ↑ ± 2.0 (95.7)	63.3 ↑ ± 2.0 (64.9)	93.2 ± 1.2 (94.0)	92.1 ↓ ± 3.2 (93.8)
5	KPT + CC + FR	84.6 ↓ ± 1.3 (85.1)	97.3 ↑ ± 0.3 (97.4)	67.3 ↑ ± 1.1 (67.7)	94.0 ↑ ± 1.2 (94.7)	92.7 ± 1.6 (93.1)
10	KPT + CC + FR	85.9 ↓ ± 1.7 (86.7)	98.1 ↑ ± 0.2 (98.2)	68.0 ± 1.1 (68.6)	93.3 ↓ ± 1.8 (93.7)	92.9 ± 1.8 (93.6)
20	KPT + CC + FR	87.3 ↑ ± 0.8 (87.6)	98.0 ↓ ± 0.4 (98.2)	69.1 ↑ ± 0.7 (69.5)	93.5 ↓ ± 1.1 (93.9)	93.1 ± 1.3 (93.5)

Table 6: Results of Contextualized Calibration and Frequency Refinement on few-shot experiments. The green up arrow ↑ means the results is higher than KPT in Table 3 and Table 5, and the red down arrow ↓ means the results is lower than KPT in Table 3 and Table 5.

Shot	Method	AG's News	DBPedia	Yahoo	Amazon	IMDB
0	KPT + ST	84.9 ↑ ± 1.0 (86.3)	81.0 ↓ ± 4.3 (85.2)	62.7 ↑ ± 1.1 (64.4)	92.8 ± 1.2 (94.7)	91.5 ↓ ± 2.8 (94.1)
1	KPT + ST	83.4 ↓ ± 3.9 (84.2)	94.0 ↑ ± 1.8 (95.8)	62.5 ↓ ± 2.3 (63.5)	93.3 ↑ ± 1.4 (94.1)	92.1 ↓ ± 3.5 (93.6)
5	KPT + ST	84.7 ↓ ± 1.8 (85.4)	97.1 ± 0.5 (97.2)	66.8 ↓ ± 1.0 (67.3)	93.3 ↓ ± 2.1 (93.8)	93.1 ↑ ± 1.4 (93.3)
10	KPT + ST	86.3 ± 1.5 (86.8)	98.0 ± 0.2 (98.1)	67.6 ↓ ± 0.9 (67.9)	94.0 ↑ ± 1.0 (94.1)	92.7 ↓ ± 1.8 (93.6)
20	KPT + ST	87.2 ↓ ± 1.1 (87.6)	97.9 ↓ ± 0.4 (98.1)	68.6 ↓ ± 0.7 (69.1)	93.5 ↑ ± 1.8 (94.0)	92.9 ↓ ± 1.2 (93.4)

Table 7: Results of restricting the expanded label word to be a single token in the PLM's vocabulary, where ST denotes "single token". The green up arrow ↑ means the results is higher than KPT in Table 3 and Table 5, and the red down arrow ↓ means the results is lower than KPT in Table 3 and Table 5.

E.5 Potential Usage without External KB.

Although KBs are ubiquitous in natural language processing, there are cases that no readily available KBs can be found for specific tasks. For these tasks, if we have enough unlabeled corpus, we can use the methods proposed by LOTClass (Meng et al., 2020) to mine potential label words from the corpus. More specifically, LOTClass (Meng et al., 2020) uses a self-supervised objective to train the PLM to extract the topic-related words from the whole unlabeled training corpus. Experiments that combine KPT with LOTClass are beyond the scope of our work, but we believe the combination of the two can be very effective.

F Datasets and Templates

We carry out experiments on three topic classification datasets: AG's News (Zhang et al., 2015), DBPedia (Lehmann et al., 2015), and Yahoo (Zhang 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 8.

Name	Type	# Class	Test Size
AG's News	Topic	4	7600
DBPedia	Topic	14	70000
Yahoo	Topic	10	60000
Amazon	Sentiment	2	10000
IMDB	Sentiment	2	25000

Table 8: 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 four manual templates that are either introduced by (Schick and Schütze, 2020a) or tailored to fit the dataset for each experimental configuration. The templates for each dataset is listed below.

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.

Yahoo. Yahoo is a topic classification dataset about the questions raised in yahoo website (Zhang et al., 2015). We use the same templates as AG's News, except that we change the word "news" into "question" in the $T_1(\mathbf{x})$:

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

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.

G Experimental Settings

We list the hyper-parameters in Table 9. Most of the hyper-parameters are the default parameters from Huggingface Transformers[7].

Hyper-parameter	Dataset	Value
truncate length	AG's News, DBPedia, Yahoo	128
truncate length	Amazon, Imdb	512
warmup steps	All	0
learning rate	All	3e-5
maximum epochs	All	5
adam epsilon	All	1e-8

Table 9: Hyper-parameter settings.

For soft verbalizer, we use a learning rate of $3e-4$ to its soft label words' embeddings to encourage a faster convergence.

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