Correcting Language Model Bias for Text Classification in True Zero-Shot Learning

Feng Zhao[1], Xianlin Wan[1], Cheng Yan[1], Chu Kiong Loo[2]
[1]Natural Language Processing and Knowledge Graph Lab,
School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2]Faculty of Computer Science & Information Technology, University of Malaya, Malaysia
{zhaof, xianlinwan, yancheng}@hust.edu.cn
ckloo.um@um.edu.my

Abstract

Combining pre-trained language models (PLMs) and manual templates is a common practice for text classification in zero-shot scenarios. However, the effect of this approach is highly volatile, ranging from random guesses to near state-of-the-art results, depending on the quality of the manual templates. In this paper, we show that this instability stems from the fact that language models tend toward predicting certain label words of text classification, and manual templates can influence this tendency. To address this, we develop a novel pipeline for annotating and filtering a few examples from unlabeled examples. Moreover, we propose a new method to measure model bias on label words that utilizes unlabeled examples as a validation set when tuning language models. Our approach does not require any pre-labeled examples. Experimental results on six text classification tasks demonstrate that the proposed approach significantly outperforms standard prompt learning in zero-shot settings, achieving up to 19.7% absolute improvement and 13.8% average improvement. More surprisingly, on IMDB and SST-2, our approach even exceeds all few-shot baselines.

Keywords: Model Bias, Prompt Learning, Text Classification, Zero-Shot Learning

1. Introduction

In recent years, fine-tuning pre-trained language models (PLMs) with task-specific data has become a standard practice for various NLP tasks (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Lewis et al., 2020; Bao et al., 2020), such as text classification (Kowsari et al., 2019), machine translation (Zhu et al., 2020), and natural language inference (Bowman et al., 2015; Williams et al., 2018). However, fine-tuning requires sufficient downstream task data to train the extra random-initialized parameters (e.g., the classification head in text classification) that it introduces. This drawback limits the application of PLMs to tasks where sufficient labeled data are unavailable and has led to research on making PLMs perform better in low-resource scenarios. Proposed by GPT-3 (Brown et al., 2020) and PET (Schick and Schütze, 2021a), prompt tuning has shown effectiveness in low-resource scenarios by incorporating human prior knowledge into the PLM's input. Prompt tuning does not need to introduce additional parameters compared to fine-tuning because it transforms the downstream task into masked language modeling, a common task in pre-training. Thus, prompt tuning utilizes the knowledge stored in PLMs in a more direct manner, which is beneficial when sufficient training data are unavailable to provide additional knowledge.

The performance of prompt learning relies on whether the PLMs can fill in the correct label word at the `[MASK]` position. However, due to the different distributions between the pre-training corpus and the task-specific data, PLMs show different propensities in predicting label words. An intuitive thought is that PLMs tend to predict label words that occur more frequently in the pre-training corpus (Zhao et al., 2021). Model bias on label words can lead to severe performance degradation on text classification in zero-shot settings since model parameters are not updated. In practice, we evaluate[1] RoBERTa-large (Liu et al., 2019) model bias on label words of AG's News[2] (Zhang et al., 2015), a four-class topic classification dataset, with a manual template. As illustrated in Figure 1(a) and Table 1(a), the model shows a much higher tendency to predict "business" than "politics", which leads to a large number of examples with the true label "politics" being incorrectly predicted as "business" (numbers underlined in Table 1(a)). Furthermore, we repeat the experiment by using another manual template. The results are shown in Figure 1(b) and Table 1(b). Surprisingly, manual template replacement greatly influences model bias on label words, which explains the dramatic performance fluctuation when changing templates in prompt tuning.

In this work, we take model bias on label words

[1]The specific method is detailed in Section 3.2.

[2]AG's News includes four categories of news: World, Sports, Business, Sci/Tech. In the experiment, we use "politics", "sports", "business" and "technology" as label words of these four categories.

LREC-COLING 2024, pages 4036–4046
20-25 May, 2024.

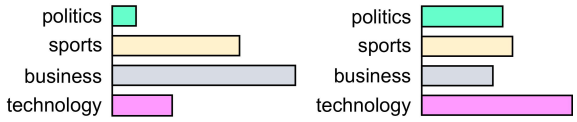
Table 1: Results of classification on AG's News with zero-shot prompt learning experiments under different templates. (a) Template: A `[MASK]` news: **x**. (b) Template: **x** This topic is about `[MASK]`. The black numbers are the number of examples being correctly classified, and the red numbers are the numbers of examples being wrongly classified. The underlined numbers cause the most accuracy loss.

Label Word	Prediction Label Word			
	politics	sports	business	technology
politics	356	310	**1217**	17
sports	2	1876	22	0
business	14	15	1767	104
technology	20	90	699	1091

(a)

Label Word	Prediction Label Word			
	politics	sports	business	technology
politics	1214	80	240	366
sports	41	1774	22	63
business	212	17	885	**786**
technology	63	26	85	1726

(b)



(a) (b)

Figure 1: Illustration of RoBERTa-large model bias on AG's News label words under different templates. (a) Template: A `[MASK]` news: **x**. (b) Template: **x** This topic is about `[MASK]`.

into consideration and develop a novel pipeline for annotating and filtering a few examples from unlabeled examples. In this way, we switch tasks from zero-shot scenarios to few-shot scenarios. Specifically, our method contains two steps: bias-based annotation and absolute probability refinement. In bias-based annotation, we randomly sample several examples from the unlabeled example set[3] and evaluate model bias on these sampled examples. For each sampled example, we reformulate it with the manual template and utilize model bias to calibrate the model prediction at the `[MASK]` position as the basis of annotation. To further improve the annotation accuracy, we propose absolute probability refinement to exclude examples with low probability on all label words. Moreover, since much prior work on few-shot learning uses a large validation set, which is unavailable in true low-data settings, to select the prompt and other model-specific hyperparameters (Perez et al., 2021), we present unlabeled validation to measure and eliminate model bias on label words while utilizing only unlabeled examples as a validation set. It is worth noting that the proposed approach does not require any pre-labeled examples, i.e., our method shows effectiveness in true zero-shot settings.

Experiments on six text classification datasets demonstrate that the proposed approach consistently outperforms standard prompt tuning in zero-shot settings, with up to 19.7% improvement and 13.8% average improvement. More surprisingly, on IMDB and SST-2, our approach yields better performance than all few-shot baselines, indicating that the proposed annotation strategy can obtain high-quality training examples from unlabeled data.

[3]We use stochastic sampling to introduce randomness to simulate the unbalanced distribution of labels in real-world scenarios.

2. Related Work

Prompt Learning. GPT-3 (Brown et al., 2020) demonstrates that large-scale PLMs can perform well in low-data scenarios by in-context learning. Specifically, instead of tuning any parameters, in-context learning concatenates the task description, a few demonstration examples and the original task input as a prompt to guide GPT-3 to predict the next word. To apply prompts on models smaller than GPT-3, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), PET (Schick and Schütze, 2021a) converts input examples into cloze questions and finetunes the model on these reformulated examples. However, manually designing good templates is laborious and requires domain knowledge. To reduce human labor in template engineering, Shin et al. (2020) proposes AUTOPROMPT to create templates automatically based on a gradient-guided search. Gao et al. (2021) proposes LM-BFF, which leverages T5 to automate the search process of templates. Searching templates over the entire vocabulary is time-consuming and suboptimal. P-Tuning (Liu et al., 2022), WARP (Hambardzumyan et al., 2021) and DART (Zhang et al., 2022) treat templates as tunable parameters and search templates in the continuous space with backpropagation. In addition, some studies have focused on verbalizer construction. KPT (Hu et al., 2022) utilizes external knowledge bases to expand and to refine the label word space of the verbalizer. DART (Zhang et al., 2022) and WARP (Hambardzumyan et al., 2021) tune the label word embeddings to achieve better representations of the labels.

Instability in Prompt Tuning. Recent work shows that the effectiveness of prompt tuning is

highly volatile, ranging from random guesses to near state-of-the-art depending on the prompt format. LAMA (Petroni et al., 2019) uses different templates to query the same information in the language models, demonstrating that the choice of templates has an impact on query accuracy. Jiang et al. (2020) reduces the instability by automatically generating diverse templates and assembling predictions when the language model uses different templates. Liu et al. (2021) shows that changing a single word in templates can drastically impact the results of prompt tuning. In prior experiments, we provide insight into how the prompt format impacts performance by influencing model bias on certain words. In addition to the template format, the choice of training data also causes instability in low-data scenarios. Schick and Schütze (2021b) finds that using different random seeds to select training data can result in significant performance fluctuations. Gao et al. (2021) incorporates training examples as demonstrations into the template and finds that the choice of demonstration examples is crucial for the final results. Zhao et al. (2021) observes that in GPT-3's input, the number and order of the demonstration examples corresponding to each label can cause accuracy to vary from near chance to near state-of-the-art. To enhance the stability of few-shot training, we propose an annotation and refinement strategy to obtain training examples with high correlation to their classes from unlabeled data.

Calibration of Prompt Tuning. The language models are usually trained with multiple large general corpora of plain text. When applying the language model to a specific downstream task, the property of the model's predicted probabilities is typically not correlated with the correctness probabilities, i.e., the language model is not calibrated for the downstream task. Jiang et al. (2021) observe that the predicted probabilities of BART, T5, GPT-2 are not calibrated on QA tasks and improve the prediction accuracy by fine-tuning and modifying the model's output. Zhao et al. (2021) consider three factors (common token bias, majority label bias and recency bias) leading to the model bias on certain answers. To reduce the influence of model bias on correctness, Zhao et al. (2021) concatenate meaningless strings into the prompt to measure model bias, and then uses model bias to adjust the model predictions on real inputs. Holtzman et al. (2021) find that language models divide the probability of the correct answer into multiple answer's synonyms when making predictions. To address this issue, Holtzman et al. (2021) modify the predicted probabilities according to answer's prior likelihood within the context. However, the above methods mainly focus on modifying the model output, and model bias still exists since model parameters are untuned. Conversely, we consider model bias while annotating examples and propose unlabeled validation to measure and eliminate model bias during training.

3. Our Approach

Our approach annotates examples with high quality from unlabeled examples based on prompt tuning and uses unlabeled examples to measure model bias on label words during training. The overview of our approach is illustrated in Figure 2. In this section, we first introduce the background of prompt tuning (Section 3.1), then present the process of annotating and refining examples (Section 3.2, 3.3), and finally, we describe how unlabeled examples can be used to eliminate model bias (Section 3.4).

3.1. Problem Definition

Let $\mathcal{M}$ be a pre-trained language model and $\mathcal{C}$ be its vocabulary. Given a text classification task $\mathcal{R}$, $\mathcal{X} = \{x_0, x_1, ..., x_n\}$ is the original input text set, where x_i denotes the i^{th} example to be classified and $\mathcal{Y} = \{y_0, y_1, ..., y_m\}$ is the label space of $\mathcal{R}$. Tackling classification tasks with prompt tuning can be roughly divided into two steps: defining a verbalizer and designing a template. A verbalizer is a function mapping y to a label word set $\mathcal{V}(y)$[4] that satisfies:

$$\mathcal{V}(y_i) \cap \mathcal{V}(y_j) = \varnothing, \ \ \forall\, 0 \leq i < j \leq m \qquad (1)$$

A template usually consists of a `[MASK]` token, a placeholder for task input, and some human-designed guiding text (e.g., A `[MASK]` news: **x**). The template can reformulate the task input to PLM's input by filling the task input text into the placeholder:

$$\tilde{x_i} = \texttt{[CLS]}\ \mathbf{x}\ \texttt{[MASK]}\ \textbf{news:}\ x_i\ \texttt{[SEP]}$$

Then $\mathcal{M}$ gives the predicted probabilities at the `[MASK]` token over the vocabulary:

$$p(w|\tilde{x_i}) = p(\texttt{[MASK]} = w|\tilde{x_i}),\ w \in \mathcal{C} \qquad (2)$$

The probability of x_i being classified into each candidate class is computed as:

$$p(y_j|\tilde{x_i}) = \sum_{w \in \mathcal{V}(y_j)} p(w|\tilde{x_i}),\ 0 \leq j \leq m \qquad (3)$$

x_i is classified into the class that obtains the highest probability.

[4]We set the size of $\mathcal{V}(y)$ to 1 for all classes in the experiments.

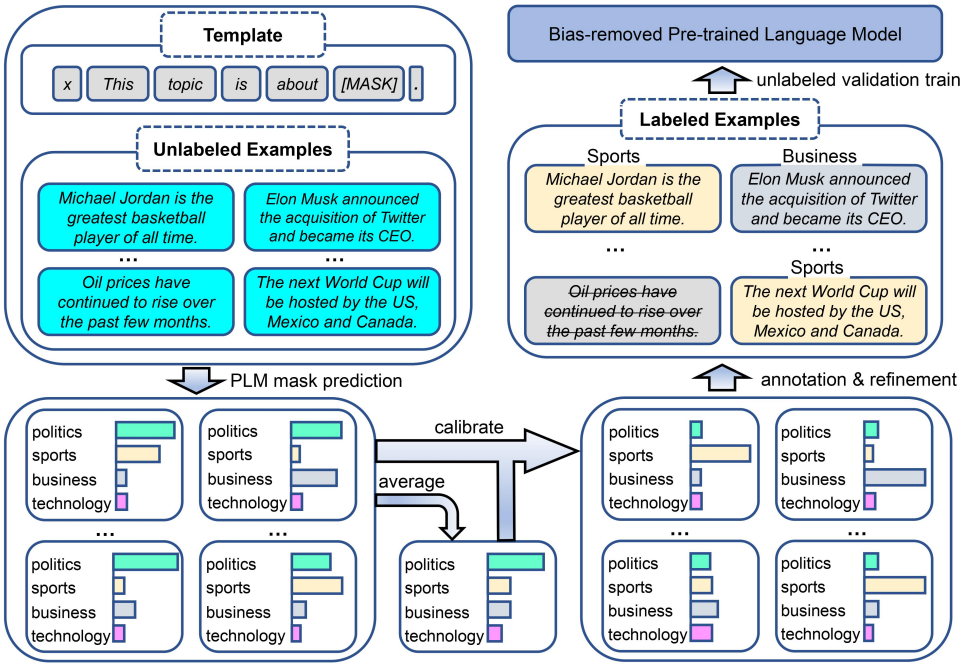


Figure 2: The framework of our approach applied to AG's News (a four-class topic classification dataset, we use "politics", "sports", "business" and "technology" as label words). We calibrate the PLM's original predictions on the unlabeled examples with model bias, and use calibrated predictions to annotate and refine training examples.

However, as shown in the previous sections, due to the model bias, certain label words consistently obtain high probabilities, regardless of the semantics of the PLM's input, which leads to many input texts being misclassified into their corresponding classes. To address this, we incorporate model bias into the data annotation process, and we propose a method to measure model bias during training with unlabeled examples.

3.2. Bias-based Annotation

The accuracy of data annotation dramatically impacts the effectiveness of subsequent model training. A critical step in bias-based annotation is to accurately measure model bias on label words in zero-shot scenarios. Given an m-class text classification task, we randomly sample $m \times k$ examples from task unlabeled data as the unlabeled validation set $\mathcal{U}$. For each example, x_i, in $\mathcal{U}$, we reformulate it into PLM's input format with the template and then calculate the probability distribution p_i over m classes. We formalize model bias on label words as follows:

$$p_b = \frac{\sum_{i=0}^{m \times k - 1} p_i}{m \times k} \tag{4}$$

As demonstrated in Figure 1, the probabilities on each label word vary greatly (e.g., in Figure 1(a), the probabilities of "politics" and "business" are 0.06 and 0.46, respectively). This bias indicates that the model is more likely to label examples as label words with high probability rather than that with low probability, which leads to a drop in annotation accuracy. To address this, we first calibrate the probability distribution of each example by element-wisely dividing the model bias:

$$\tilde{p_i}(j) = \frac{p_i(j)}{p_b(j)},\ 0 \le i < m \times k, 0 \le j < m \tag{5}$$

where $p(j)$ is the j^{th} element of p. Then, we classify these examples according to the highest probability in the calibrated probability distribution. Inside each class, we choose the top-n examples with the highest probability as the training data $\mathcal{T}$:

$$\mathcal{T} = \bigcup_{i=1}^{m} \{top\text{-}n[\mathcal{X}_i]\} \tag{6}$$

where $\mathcal{X}_i$ represents the examples categorized to class i.

3.3. Absolute Probability Refinement

We call the probability of label words over the vocabulary *absolute probability* and that over each class *relative probability*. The annotation strategy proposed in Section 3.2 is based on the relative probabilities of the label words in both calibration and top-n selection. However, solely relying on relative probability may incur mistakes when annotating examples with low absolute probabilities for all

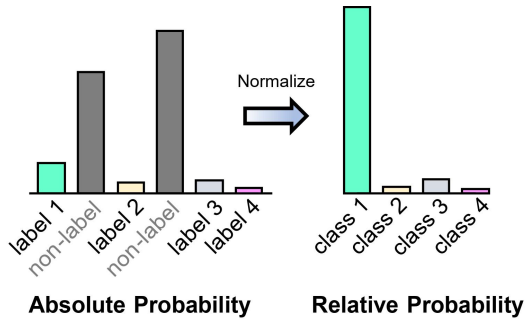


Figure 3: Left: The model's prediction distribution on the vocabulary for an example not fitting any class. Right: The relative probability distribution of the example over each class after normalization.

label words. As shown in Figure 3, although the example does not fit into any class, the relative probability of class 1 is close to 1 after normalization, putting it in the forward position in subsequent top-n selection. To improve annotation accuracy, we propose *absolute probability refinement*, which introduces an additional condition to top-n selection. Specifically, we set a probability threshold for each class. During top-n selection, we filter out examples with absolute probability lower than the probability threshold of their class. By incorporating absolute probability into the annotation, we enhance the correlation between the selected training examples and their classes.

3.4. Unlabeled Validation

Previous work has shown that prompt engineering is a crucial step in prompt learning. However, the effectiveness of prompt learning varies considerably even when modifying a word in a template without changing the semantics. Some research has used a large validation set to determine the best template and when to stop training, which is not applicable in true zero-shot scenarios. In this work, instead of evaluating the model's accuracy on a validation set, we measure whether the model bias distribution on the unlabeled validation set $\mathcal{U}$ is balanced over label words since the training objective is to eliminate model bias. Ideally, the model bias on label words should be nearly uniformly distributed:

$$\begin{cases} p_{avg}(i) & = \frac{1}{m} + \sigma_i\,, \quad 0 \le i < m \\ \sum_{i=0}^{m-1} \sigma_i & = 0 \end{cases} \tag{7}$$

where σ_i is a randomly generated small number that represents noise. Therefore, we use the distance between p_b and p_{avg} to represent model bias on label words during training:

$$d = \sqrt{\sum_{i=0}^{m-1} (p_b(i) - p_{avg}(i))^2} \tag{8}$$

4. Experiments

We conduct experiments on six text classification datasets to show the effectiveness of our approach. In this section, we first introduce statistics for the six datasets, the experimental settings we used, and the baselines for comparison with our approach. Then, we present our main results and provide possible insights into our method.

4.1. Dataset Statistics

We conduct experiments on six popular text classification datasets, including three topic classification datasets: DBPedia (Lehmann et al., 2015), AG's News (Zhang et al., 2015) and Yahoo (Zhang et al., 2015), and three sentiment classification datasets: IMDB (Maas et al., 2011), Amazon (McAuley and Leskovec, 2013) and SST-2 (Socher et al., 2013). DBPedia, AG's News and Yahoo contain 14, 4 and 10 categories, respectively. All three sentiment classification tasks have two polarities, i.e., positive and negative. The statistics of the datasets are shown in Table 2.

Table 2: The statistics of the datasets used in the experiments.

Dataset	Type	# Class	# Test Example
DBPedia	Topic	14	70000
IMDB	Sentiment	2	25000
Amazon	Sentiment	2	10000
SST-2	Sentiment	2	872
AG's News	Topic	4	7600
Yahoo	Topic	10	60000

4.2. Experimental Settings

Our experiments are built on Pytorch. We use RoBERTa-large (Liu et al., 2019) as the base model for all experiments and report the accuracy. For prompt-based methods, we follow the setup of KPT (Hu et al., 2022) with four manual templates and repeat the experiments with five different random seeds for each template, which significantly eliminates the randomness in experiments and makes our results convincing. For the fine-tuning method, we use the same five seeds as prompt-based methods for a fair comparison. For bias-based annotation, we randomly select $m \times 200$ examples for an m-class task from the task unlabeled data as the unlabeled validation set $\mathcal{U}$ and then annotate five examples per class as training

data from $\mathcal{U}$. For absolute probability refinement, we evaluate the absolute probability of all examples in each class and take the median as the probability threshold. We train the model for five epochs with the learning rate set to 3e-5 and the batch size set to 4 in all experiments. We evaluate model bias on the unlabeled validation set every epoch and choose the least biased checkpoint to test.

4.3. Baselines

To put our results in perspective, we compared our approach with the following baselines. Since the proposed approach is under true zero-shot settings, two zero-shot learning methods are included in the compared baselines. In addition, given that our method is based on few-shot annotation, we compare it with three few-shot learning methods to demonstrate the precision of the annotation method.

Fine-tuning (FT). The fine-tuning method adds a random-initialized classification head on top of the PLM. The classification head takes the last hidden states of the `[CLS]` token as input and makes predictions. Fine-tuning updates the model parameters and the classification head parameters during training.

Prompt-tuning (PT). Proposed by GPT-3 (Brown et al., 2020) and PET (Schick and Schütze, 2021a), the prompt-tuning method converts input examples into cloze questions and maps PLM's prediction words on the `[MASK]` token to classes via the verbalizer. For a fair comparison, all prompt-based methods use the same templates and verbalizers.

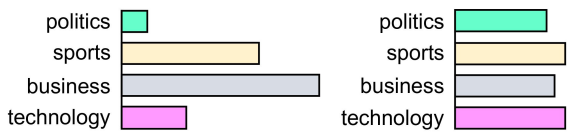
Contextual Calibration (CC). Contextual calibration is proposed by Zhao et al. (2021). They first evaluate model bias on label words by concatenating a content-free text at the end of the prompt as input to GPT-3. Then they calibrate the model predictions by element-wisely dividing model bias.

For our method, we conduct ablation experiments to evaluate the effectiveness of each module. -UV, -APR and -BA denote the absence of unlabeled validation, absolute probability refinement and bias-based annotation, respectively. In addition, we incorporate our proposed unlabeled validation into few-shot prompt-tuning to further illustrate its effect.

4.4. Main Results

As shown in Table 3, our approach outperforms zero-shot PT by a large margin (on average +13.8%), especially on DBPedia, Amazon and SST-2, with improvements up to 19.7%, 18.3% and 19.6%, respectively. Compared to zero-shot PT+CC, our approach also consistently obtains better performance with an improvement of 10.2% on DBPedia, 7.9% on Amazon, and an average improvement of 5.2% on all datasets. Thus, the proposed approach exceeds all baselines in true zero-shot settings. Moreover, on IMDB and SST-2, our approach even outperforms all baselines in few-shot settings. In this regard, our conjecture is that the examples labeled by our annotation and refinement algorithm are not only correct but also more correlated with the classes compared with the randomly selected examples used in few-shot methods. Comparison between few-shot PT and few-shot PT+UV demonstrates that unlabeled validation can boost the effect of prompt tuning in few-shot settings without large validation sets. In terms of stability, the standard deviation of our method is smaller than other baselines in most cases, which indicates that our method can maintain good performance with different templates and can therefore reduce human labor in template engineering. Our insight on this is that different templates cause performance fluctuations by impacting model bias on label words, while our approach can greatly eliminate model bias during training.

In ablation experiments, we observe that the performance of our approach decreases as we eliminate UV, APR, and BA in sequence, demonstrating the effectiveness of each module. Furthermore, we observe that the absence of bias-based annotation causes the most performance loss, especially in tasks with more classes, such as DBPedia and Yahoo. We find that without bias-based annotation, the accuracy of labeled examples drops considerably, and in some situations, no examples are annotated as classes corresponding to low-probability label words due to model bias.



(a) before calibration (b) after calibration

Figure 4: Model bias on four label words of AG's News before and after calibration.

5. Analysis

5.1. Model Bias after Calibration

The performance of our approach shows a substantial improvement compared to zero-shot prompt tuning. To further demonstrate that the improvement stems from calibrating model bias on label words, we first use the same dataset and template as in Figure 1(a) and measure model

Table 3: Results on classification tasks. †: the full training set is used; ‡: use $K = 5$ (per class) for few-shot experiments; otherwise, no pre-labeled examples are used. For prompt-based method, we report the mean and the standard deviation performance of four templates on five random seeds. For fine-tuning, we report performance on five random seeds. Majority: majority class; FT: fine-tuning; PT: prompt tuning. CC means contextual calibration; UV, APR and BA means unlabeled validation, absolute probability refinement and bias-based annotation, respectively. **bold**: the best performance among zero-shot methods; underline: the best.

Method	DBPedia	IMDB	Amazon	SST-2	AG's News	Yahoo
Majority†	7.1	50.0	50.0	50.9	25.0	10.0
Few-shot FT‡	94.9 ± 1.9	64.4 ± 6.3	61.6 ± 9.4	54.3 ± 4.3	72.3 ± 7.7	20.6 ± 6.5
Few-shot PT‡	96.4 ± 0.7	82.1 ± 13.3	90.7 ± 5.2	76.8 ± 13.3	82.2 ± 3.2	61.1 ± 1.7
Few-shot PT+UV‡	96.4 ± 0.9	83.9 ± 14.4	94.0 ± 1.1	75.9 ± 12.9	84.7 ± 2.2	61.5 ± 1.7
Zero-shot PT	68.0 ± 3.4	84.3 ± 12.4	75.5 ± 11.5	68.3 ± 13.4	75.9 ± 5.1	47.5 ± 7.0
Zero-shot PT+CC	77.5 ± 6.2	89.3 ± 5.0	85.9 ± 3.9	82.4 ± 5.2	79.6 ± 2.5	56.4 ± 2.5
Ours	**87.7 ± 5.9**	**92.2 ± 1.5**	**93.8 ± 1.3**	**87.9 ± 3.1**	**82.0 ± 2.6**	**58.4 ± 1.9**
- UV	86.7 ± 5.4	89.2 ± 4.3	91.0 ± 2.5	84.8 ± 4.9	80.2 ± 2.9	57.6 ± 2.8
- UV - APR	86.3 ± 5.7	88.0 ± 4.8	90.9 ± 3.5	85.0 ± 4.3	79.4 ± 3.2	57.2 ± 2.8
- UV - APR - BA	69.7 ± 5.9	86.8 ± 3.9	87.9 ± 5.8	80.7 ± 6.7	72.6 ± 4.2	42.8 ± 2.6

Table 4: Results of classification on AG's News using the same template as in Table 1(a).

Label Word	Prediction Label Word			
	politics	sports	business	technology
politics	1420	116	252	112
sports	10	1873	11	6
business	46	18	1491	345
technology	69	80	102	1649

Table 5: The average accuracy (%) of our approach on AG's News under different sizes of unlabeled data and training examples.

# Unlabeled Data	K Training Examples			
	5	10	15	20
200	82.0	76.8	71.9	71.3
400	82.3	83.1	78.6	73.7
600	81.8	82.9	83.5	80.0
800	82.0	82.3	83.2	82.8

bias after training. The results are shown in Figure 4. Compared to the high probability of "business" and the low probability of "politics" before calibration, the probability distribution is more uniform across label words after training the model with our approach. Then we tabulated the model's predictions on the test set on each label word, as shown in Table 4. We find that the number of examples with the label word "politics", which are incorrectly predicted as "business", drops from 1217 (underlined in Table 1(a)) to 252, contributing the most to accuracy improvement. Thus, the proposed approach can eliminate model bias on the label words and can reduce the number of examples with low-probability label words being misclassified as classes with high-probability label words.

5.2. Analysis of Unlabeled Data

In bias-based annotation, we find that the size of unlabeled data has an impact on the precision of measuring model bias, which consequently affects the accuracy of data annotation and the size of training set. In our previous experiments, we use $m \times 200$ examples as the unlabeled data for an m-class task and annotate five examples for each class as training data. However, the size of unlabeled data is uncertain in real-world scenarios. Thus we conduct experiments on AG's News to demonstrate the influence on the performance of our approach when the size of unlabeled data and training set are changed. As demonstrated in Table 5, the best performance of our method is achieved when ratio of the unlabeled data size to the training data size is 40. When the ratio decreases, the accuracy drops considerably due to the increase in the number of incorrectly annotated training examples. Figure 5 shows that as the ratio of N (# unlabeled examples per class) to K (# training examples per class) decreases, the number of mislabeled examples in the training set increases rapidly, which leads to a decrease in the accuracy of the model after training.

[5]XLNet-large is a decoder-only model, thus the `[MASK]` in the template must be placed at the end. BERT-large and ALBERT-xxlarge are encoder-only models, which have no restriction on the position of the `[MASK]` in the template. To demonstrate that the model bias does not vanish when changing templates, we use a different template for BERT-large and ALBERT-xxlarge.

Table 6: Model bias on AG's News before and after calibration. The template used in XLNet-large: **x** This topic is about [MASK]. The template used in BERT-large and ALBERT-xxlarge: A [MASK] news: **x**[5].

LLMs	Model Bias			
	politics	sports	business	technology
XLNet-large	0.28	0.11	0.21	0.40
BERT-large	0.30	0.42	0.10	0.18
ALBERT-xxlarge	0.19	0.33	0.21	0.27

(a) before calibration

LLMs	Model Bias			
	politics	sports	business	technology
XLNet-large	0.24	0.23	0.26	0.27
BERT-large	0.25	0.28	0.23	0.24
ALBERT-xxlarge	0.26	0.25	0.25	0.24

(b) after calibration

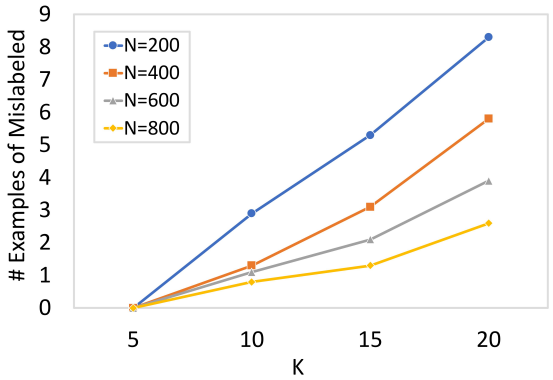


Figure 5: A lower ratio of N (# unlabeled examples per class) to K (# training examples per class) results in an increase in mislabeled training examples.

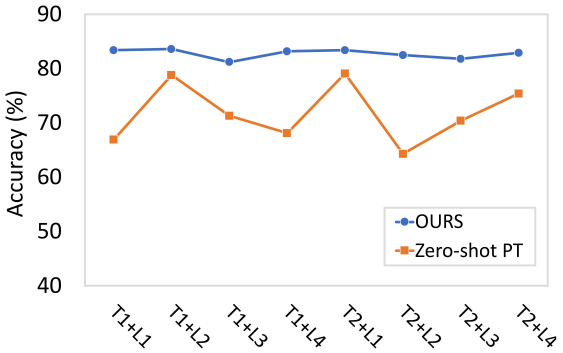


Figure 6: Accuracy of our approach and Zero-shot PT on AG's News using different combinations of templates and label word sets.

5.3. Sensitivity to Templates and Label Words

As shown in Figure 1 and Table 1, the effect of prompt tuning is sensitive to different templates, which makes it hard to design templates manually. One advantage of our approach is that it can calibrate model bias on label words and thus reduce the sensitivity to prompts. We conduct a case study on AG's News using two templates[6] with four label word sets[7]. As shown in Figure 6, the accuracy of our approach maintains stability while consistently outperforming Zero-shot PT.

5.4. Model Bias in Other LLMs

We use RoBERTa-large as the backbone in previous experiments. To verify whether model bias exists in other large language models, we measure the model bias of XLNet-large (Yang et al., 2019), BERT-large (Devlin et al., 2019), and ALBERT-xxlarge (Lan et al., 2020) on the label words of AG's News. As shown in Table 6(a), model bias exists in all three models and different models show various distributions of model bias. Moreover, after correcting the model bias with our approach, we measure the model bias again on AG's News using the same unlabeled data and template. As shown in Table 6(b), the model bias of all three models is nearly uniformly distributed, which demonstrates the effectiveness of our method.

Table 7: The impact of model bias in XLNet-large on predicting the next word.

Next Word	Prediction of Next Word			
	politics	sports	business	technology
politics	1651	6	52	191
sports	255	623	118	904
business	129	27	1343	401
technology	54	2	16	1828

5.5. Applicability beyond Classification

For text classification, model bias on label words can directly affect the classification accuracy. For other NLP tasks, such as text generation, we argue that model bias similarly affects model performance. As demonstrated in Table 6(a), XLNet-large shows large model bias on the label words of AG's News, which affects the probability of predicting the next word. Table 7 demonstrates that model bias leads to incorrect prediction of the next word.

[6]Template 1: A [MASK] news: **x**. Template 2: **x** This topic is about [MASK].

[7]Label word set 1: politics, sports, business, technology. Label word set 2: country, athletics, commerce and science. Label word set 3: politics, sports, commerce and science. Label word set 4: country, athletics, business, technology.

6. Conclusion

In this paper, we first show that model bias on label words can impact the performance of prompt learning and that different templates lead to instability in prompt learning by affecting the model bias. Then, we propose a data annotation and filtering method that incorporates model bias in true zero-shot settings. Finally, we use unlabeled data to select the least biased model during training. The experiments demonstrate that our approach can calibrate model bias on label words and thus can improve the accuracy of text classification tasks. In the future, we intend to incorporate continuous prompts and multi-verbalizers into our approach to further reduce the impact of model bias on prompt learning.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2023YFF0905503, National Natural Science Foundation of China under Grants No.62072203, and Malaysia MOHE FRGS Funding No. FRGS2023-1.

7. Bibliographical References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 642–652. PMLR.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transac-*

tions of the Association for Computational Linguistics, 9:962–977.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

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. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/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 L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 165–172. ACM.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34:11054–11070.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 2339–2352, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.