Fine-tuning after Prompting: an Explainable Way for Classification

Zezhong Wang[1,2*] , **Luyao Ye**[3*] , **Hongru Wang**[1,2],
Boyang Xue[1,2], **Yiming Du**[1,2], **Bin Liang**[1,2], **Kam-Fai Wong**[1,2]
[1]The Chinese University of Hong Kong, Hong Kong, China
[2]MoE Key Laboratory of High Confidence Software Technologies, China
[3]Central China Normal University, Wuhan, China
zzwang@se.cuhk.edu.hk

Abstract

Prompting is an alternative approach for utilizing pre-trained language models (PLMs) in classification tasks. In contrast to fine-tuning, prompting is more understandable for humans because it utilizes natural language to interact with the PLM, but it often falls short in terms of accuracy. While current research primarily focuses on enhancing the performance of prompting methods to compete with fine-tuning, we believe that these two approaches are not mutually exclusive, each having its strengths and weaknesses. In our study, we depart from the competitive view of prompting versus fine-tuning and instead combine them, introducing a novel method called F&P. This approach enables us to harness the advantages of **F**ine-tuning for accuracy and the explainability of **P**rompting simultaneously. Specifically, we reformulate the sample into a prompt and subsequently fine-tune a linear classifier on top of the PLM. Following this, we extract verbalizers according to the weight of this classifier. During the inference phase, we reformulate the sample in the same way and query the PLM. The PLM generates a word, which is then subject to a dictionary lookup by the verbalizer to obtain the prediction. Experiments show that keeping only 30 keywords for each class can achieve comparable performance as fine-tuning. On the other hand, both the prompt and verbalizers are constructed in natural language, making them fully understandable to humans. Hence, the F&P method offers an effective and transparent way to employ a PLM for classification tasks.

1 Introduction

Prompting (Heinzerling and Inui, 2021) is a novel method for adapting pre-trained language models (PLMs) to downstream classification tasks (Brown et al., 2020; Zhao et al., 2024). Generally, a prompt typically consists of a sample, a task description, and a reserved blank. PLM is required to generate an appropriate word to fill in this blank based on the task description and the sample. A verbalizer then assigns a class to this word, finalizing the sample's classification. For example,

I like this movie. The sentiment is ____.

is a manual prompt designed for sentiment analysis. A typical verbalizer uses a lookup table to determine the class to which the predicted word should belong (Schick and Schütze, 2021; Hu et al., 2021; Webson and Pavlick, 2021; Ding et al., 2022). In this manner, a classification task is transformed into a language modeling task, aligning with the pre-training tasks of PLMs.

Compared with fine-tuning, prompting methods are more transparent to humans as the prompt consists of real words and is more explainable than a classifier with numerous parameters. However, prompting methods exhibit a lower performance than fine-tuning (Shin et al., 2020; Jiang et al., 2020). Because of the context sensitivity inherent to PLMs, their responses to identical queries exhibit inconsistencies when prompted in varying ways. Simply altering the wording of prompts, or even making minor lexical adjustments, can result in performance variations of up to 20% (Jiang et al., 2020). To this end, a series of studies (Liu et al., 2021; Zhong et al., 2021; Qin and Eisner, 2021; Wang et al., 2022; Li and Liang, 2021; Wang et al., 2023; Li et al., 2023) delved into methods for formulating effective prompts. They believed that prompts are not necessarily composed of real words and proposed a novel approach called "prompt tuning," wherein a set of k trainable vectors is employed as prompts, rather than conventional natural language, e.g.,

I like this movie. $v_1, \cdots, v_k$ ____.

These methods greatly enhance the capabilities of prompts, yielding performance comparable to or

[*] Equal contribution.

even surpassing that of fine-tuning. However, it is noteworthy that prompts become less explainable for humans. Despite the absence of explicit research on the connection between explainability and performance, current efforts inadvertently prioritize performance over explainability when developing capable prompts. We believe that the relationship between explainability and performance is not mutually exclusive. It is feasible to enhance prompt performance while simultaneously taking into account their explainability. In this work, we depart from the competition paradigm between prompting and fine-tuning. Instead, we integrate both techniques and propose a novel method F&P that attains performance on par with **F**ine-tuning, while preserving the outstanding explainability inherent to **P**rompting methods.

Specifically, referring to the prompting method, we create a task description for each classification task and leave a blank space for the PLM to make predictions. We concatenate such a task description at the end of each sample, forming a prompt. Next, we refer to the fine-tuning, by adding a linear layer on top of the PLM to classify its output. It is worth mentioning that traditional fine-tuning methods often replace the Language Model Head with a linear layer, whereas we add an additional linear layer on top of the Language Model Head. Therefore, the linear layer classifies the word distribution predicted by the PLM rather than word embeddings. Furthermore, in contrast to classifying sentence representations, such as `[CLS]`, we classify the word distribution output from the blank space in the model. After the fine-tuning, the weights of the linear layer represent the significance of words for each class. To create a verbalizer, we sort all words in the vocabulary based on these weights and select the top-k words for each class. Then we remove the linear layer. During inference, given the new sample, we construct the prompt in the same way and input it into the PLM. The PLM's predicted word is then associated with a class based on the verbalizer. In this approach, we replace the classifier with a prompt and a verbalizer, yielding two key advantages. Firstly, both the prompt and verbalizer employ real, easily understandable words, making the classification process transparent to humans. This contrasts with the use of complex classifiers, which often obscure the classification process. Secondly, this approach avoids introducing additional parameters to the PLM, allowing us to maintain the original PLM size. Consequently, it can be applied to prob the linguistic knowledge embedded in the PLM, a crucial technique to explain the PLM (Tenney et al., 2019; Li et al., 2022).

2 Related Work

Fine-tuning represents the predominant method for customizing PLMs to specific downstream tasks. However, these tasks often diverge significantly from the cloze test used during the PLM's pre-training phase. For instance, RoBERTa (Liu et al., 2019) demonstrates proficiency across various tasks such as text classification, and sequence labeling. However, its pre-training task is a cloze test. The disparity between pre-training tasks and downstream applications is believed by researchers to hinder the optimal utilization of PLMs' knowledge (Han et al., 2021). This gap poses challenges, notably the propensity for PLMs to exhibit overfitting on limited training samples post fine-tuning, particularly when data availability is constrained. Therefore, addressing this gap is crucial to fully harnessing the potential of PLMs across diverse applications and ensuring robust performance in practical scenarios.

Prompt-based methods have been introduced as a strategic bridge between pre-training and fine-tuning stages in NLP. According to Petroni et al. (2019), these methods leverage the relational knowledge inherently encoded within PLMs, thereby demonstrating their efficacy in various tasks. Additionally, Brown et al. (2020) substantiated that the expansive knowledge encoded within large-scale PLMs is substantial enough to execute tasks effectively without necessitating parameter tuning. Furthermore, these methods enhance the usability of PLMs across different tasks by appending supplementary descriptions and examples in a cloze-style format, aligning each downstream task consistently with the structure of the pre-training tasks. This standardization not only facilitates smoother transitions between stages but also optimizes task performance. Recent studies have underscored the competitive advantages of prompt-based methods, showing that they can achieve comparable or superior performance compared to traditional fine-tuning approaches (Gao et al., 2021; Qin and Eisner, 2021; Zhong et al., 2021; Zhu et al., 2022; Li et al., 2021; Chen et al., 2022; Wang et al., 2023). Moreover, they have demonstrated remarkable efficacy in scenarios requiring minimal training data, such as few-shot or zero-shot settings. This adapt-

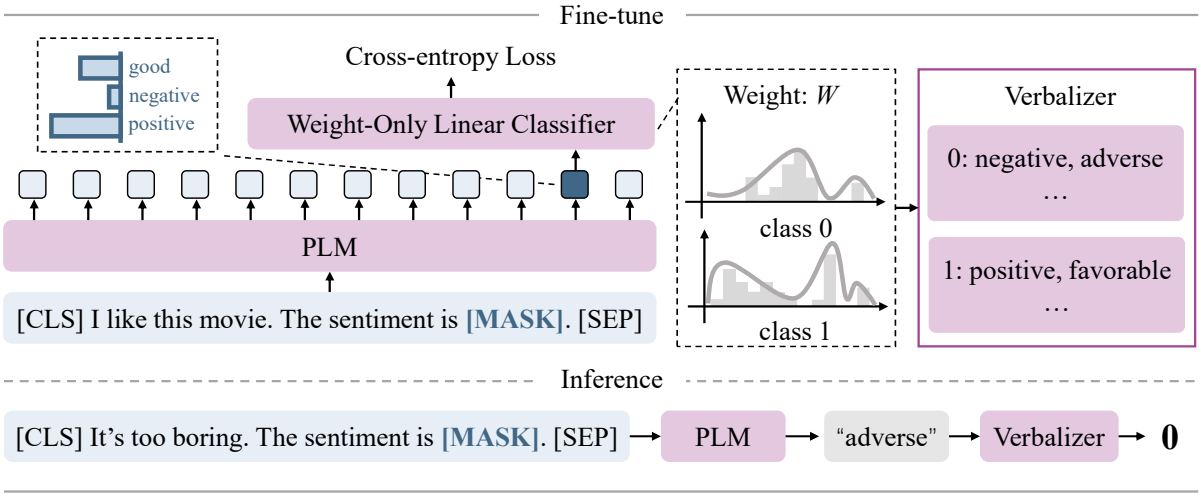

Figure 1: The upper part illustrates the process of fine-tuning the whole model and constructing the verbalizer from the classifier's weight. The lower part shows the inference process with a tuned PLM and the verbalizer.

ability underscores their potential to significantly advance the field of NLP by making efficient use of pre-existing model knowledge (Schick and Schütze, 2021; Puri and Catanzaro, 2019; Schick et al., 2020; Zhang et al., 2021; Ben-David et al., 2022).

3 Methodology

Figure 1 illustrates the overview of F&P. A prompt p is composed of three parts, including an input x, a task description with k tokens $t_1, \cdots, t_k$, and a symbol of mask, i.e., $p = [x, t_1, \cdots, t_k, \text{[MASK]}]$. Fed the prompt p, the PLM $\mathcal{F}(\cdot)$ predicts the word distribution for [MASK]:

$$\mathcal{F}(p) = P([\text{MASK}] = t_i|p), t_i \in \mathcal{V} \quad (1)$$

where $\mathcal{V}$ is the vocabulary that contains n unique words t_i. In practice, the PLM's output is in the format of a vector, i.e., $\mathcal{F}(p) \in \mathbb{R}^n$. We add a weight-only linear classifier on top of the PLM to project $\mathcal{F}(p)$ into C classes, i.e.,

$$y = W^T \text{softmax}(\mathcal{F}(p)) \quad (2)$$

where $W \in \mathbb{R}^{n \times C}$. We use the cross-entropy loss as the objective and fine-tune the model until converge. After fine-tuning, each column of the classifier's weight, i.e., $W_i^T \in \mathbb{R}^n$, can represent how significant a word is to the class i. We rank and select top-k words from the vocabulary with the highest weight in W_i^T as the mapping to the class i, i.e.,

$$\mathcal{M}_i : i \leftarrow \{t_j | j \in \underset{j}{\text{top-k}}([W_{ij}^T]_{1 \le j \le n})\} \quad (3)$$

where t_j is the j-th token in the PLM's vocabulary. We gather all mappings of classes to construct a lookup table as the verbalizer $\mathcal{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_C\}$.

In the inference, the input is wrapped into a prompt $\hat{p}$ in the same way and processed by the PLM following the equation 1. The token in the verbalizer with the largest probability is the predicted word, i.e., $t^* = \arg\max P([\text{MASK}] = t_i|\hat{p}), t_i \in \mathcal{M}$. The final prediction is made by looking up the verbalizer, $\mathcal{M}(t^*)$.

4 Experiments

4.1 Experiment Setting

4.1.1 Datasets

We conducted experiments on two benchmarks, GLUE(Wang et al., 2018) and CLUE (Xu et al., 2020).

- **General Language Understanding Evaluation (GLUE)** benchmark comprises nine natural language understanding tasks. These include single-sentence tasks like CoLA and SST-2, similarity and paraphrasing tasks such as MRPC, STS-B, and QQP, and natural language inference tasks including MNLI, QNLI, RTE, and WNLI.

- **Chinese Language Understanding Evaluation (CLUE)** is a community-driven, open-ended project that combines nine tasks, covering well-established single-sentence and sentence-pair classification tasks, as well as

LLM	Checkpoints
BERT-base	`bert-base-cased`
BERT-large	`bert-large-cased`
RoBERTa-base	`roberta-base`
OpenAI GPT	`openai-gpt`
BERT-wwm-ext-base	`chinese-bert-wwm-ext`
RoBERTa-wwm-ext-base	`chinese-roberta-wwm-ext`
RoBERTa-wwm-ext-large	`chinese-roberta-wwm-ext-large`

Table 1: PLMs involved in the experiments and the corresponding checkpoints.

machine reading comprehension, all based on original Chinese text.

The dataset split schema adheres to the same configuration as the benchmark.

4.1.2 PLMs

All experiments are conducted with four PLMs including BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), `OpenAI GPT` (Radford et al., 2018), and the Chinese version of BERT and RoBERTa (Cui et al., 2020). We provide the detailed version of checkpoints in Table 1.

4.1.3 Baseline Methods

We compare three methods to tune PLMs:

- **Fine-tuning (FT)** refers to the process of replacing the Language Model Head of a the PLM with a linear classifier and subsequently updating the entire model. The input for this linear classifier is the sentence representation generated by the PLM.
- **Fine-tuning and Prompting (F&P)** is our method presented in the section 3.
- **Fine-tuning and AUTOPROMPT (F&AP).** AUTOPROMPT (Shin et al., 2020) is a method to search prompts automatically. We consider it an enhancement tool for identifying high-performing prompts. We employ it to discover six trigger words, denoted as $t_1, \cdots, t_6$, within the training dataset to replace the manual prompt. Subsequently, we repeat the same procedures as described in F&P.

4.1.4 Hyperparameter

For all of the experiment, We use Adam (Kingma and Ba, 2014) as the optimizer with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-6$. Besides that, we consider a hyperparameter grid search for each task, with weight decay $\in \{1e-5, 1e-4, 1e-3\}$ and learning rates $\in \{1e-5, 2e-5, 3e-5\}$, with an exponential warmup for the first 8% of steps followed by a linear decay to 0.

4.2 Main Results

4.2.1 The verbalizer can be regarded as a classifier after denoising.

F&P does not introduce any extra parameters to the PLM. But from Table 2, it achieves performance that is comparable to, or even superior to, fine-tuning, which involves the incorporation of an extra linear classifier.

For example, when using BERT-base, employing just 5 words per class can effectively substitute a classifier that typically requires over 22 million parameters. Despite this reduction in complexity, the model experiences only a marginal decrease in performance, approximately 0.22%. Interestingly, in certain scenarios, there is even a noticeable improvement; for instance, the accuracy of BERT-large on MNLI-mm rises from 85.25% to 86.82%. This phenomenon can be explained from a denoising standpoint. In contrast to a conventional weight-only classifier, the verbalizer, by focusing on a select set of words, tends to omit information associated with words carrying lower weights. Moreover, it simplifies the prediction process by treating all selected words equally in determining the outcome, thereby potentially enhancing clarity and reducing noise in the decision-making process. This selective attention mechanism not only streamlines the model but also serves as an effective denoising filter, enhancing overall performance in certain tasks.

We fine-tune the BERT-base model on the SST-2 dataset using the F&P method. Through this process, we extracted the weights of the linear classifier and proceeded to visualize the difference between the weights assigned to the two classes, specifically denoted as $W_1 - W_0$. Moreover, to underscore the efficacy of our approach, we also visualized the verbalizer generated by F&P in a manner similar to the weight comparison, demonstrating its effectiveness in enhancing classification accuracy.

In Figure 2, the left side appears disorderly, with words displaying uniform weights and lacking meaningful differentiation. These words are primarily noise rather than informative features. Conversely, the right side shows the word weights after ranking and selection, resulting in the removal of most of the words. Although this process might

PLM	Method	CoLA	SST-2	MRPC	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	Avg.
BERT-base	FT.	57.35	93.42	90.03	84.36	84.54	83.22	90.81	70.86	74.41	81.00
	F&P.$_{(5)}$	57.52	93.63	89.18	83.20	83.12	82.26	91.25	71.82	75.01	80.78
	F&AP.$_{(5)}$	62.52	94.10	92.52	88.53	84.18	83.35	92.91	73.14	76.75	83.11 (+2.11)
BERT-large	FT.	61.90	93.62	88.57	85.50	86.13	85.25	93.65	71.16	75.31	82.34
	F&P.$_{(10)}$	62.74	93.27	89.34	84.14	86.04	87.31	93.92	72.53	77.04	82.93
	F&AP.$_{(10)}$	63.92	94.50	90.97	87.44	86.27	86.82	95.13	73.82	76.77	83.96 (+1.62)
RoBERTa-base	FT.	68.15	95.71	91.15	89.12	90.11	90.02	94.36	85.92	90.00	88.28
	F&P.$_{(30)}$	67.63	94.95	90.33	88.20	89.40	89.18	93.10	84.90	89.53	87.47
	F&AP.$_{(30)}$	68.42	96.62	91.51	91.05	90.25	91.65	95.82	86.66	91.47	89.27 (+0.99)
OpenAI GPT	FT.	37.18	93.50	87.55	69.17	81.65	80.20	84.35	63.74	72.73	74.45
	F&P.$_{(5)}$	46.17	93.46	89.14	76.60	81.23	81.42	83.70	66.28	73.08	76.79
	F&AP.$_{(5)}$	50.12	93.90	90.27	77.84	82.54	82.15	84.53	67.81	73.88	78.12 (+3.67)

Table 2: Results on the development set of GLUE benchmark. F1 score (%) is the metric used for MRPC and QQP, Matthew's Correlation for CoLA, and Accuracy (%) for the other tasks. The number in the bracket indict the number of words selected for each class for the verbalizer. , e.g., $_{[10]}$ means select top-10 words from each class. The number in red represents the improvements of F&AP over the fine-tuning.

involve some loss of information obtained from the training dataset, it significantly enhances the verbalizer's overall generality and effectiveness.

4.2.2 Prompts improve the distinctiveness of the model's output.

The performance analysis of F&AP revealed an improvement of approximately 2% compared to fine-tuning alone, suggesting that fine-tuning procedures may not fully exploit the inherent capabilities of PLMs. The inclusion of a prompt in the form of *"The sentence is [MASK]."* serves to constrain the output range of the PLM by introducing a fixed component within the context. This prompt requires the PLM's predictions to align with the given context, thereby encouraging the model to emphasize specific attributes crucial to the task during the fine-tuning process. This approach offers a method to enhance classification performance through context adjustment, complementing rather than contradicting traditional fine-tuning methodologies.

4.2.3 Verification on Chinese Dataset

We also validated the F&P method on CLUE, a Chinese dataset. The experimental results are shown in Table 3. Overall, the F&P method still outperformed traditional fine-tuning, with slight improvements across multiple models and tasks. This confirms the effectiveness of our approach not only in English but also in Chinese tasks.

However, the improvements on the Chinese dataset were not as significant as those on the English dataset. We attribute this mainly to suboptimal prompt designs for Chinese tasks. Since the AUTOPROMPT method was originally proposed for English data, although there is no evidence suggesting it only works for English, this experiment shows limited improvement on Chinese datasets. In the future, we will further tune this method to find optimal Chinese prompts for each task.

4.3 Explain the Verbalizer

Traditional classifiers typically involve a multitude of parameters whose complex interactions can obscure the decision-making process, even when the operations involved are purely linear. In contrast, prompting the PLM, mapping and aligning their outputs with specific classes through a verbalizer offers a stark contrast in transparency for human observers. As discussed by Molnar (2020), explainability refers to the degree to which a person can reliably anticipate the model's predictions. In this framework, the consistency of the verbalizer becomes paramount, ensuring a cohesive semantic alignment with the assigned class labels. For instance, if the term "favorable" is linked with the "Negative" class, such discrepancies highlight a breakdown in the verbalizer's coherence with human comprehension, thereby compromising interpretability.

4.3.1 Consistency Test Between Verbalizers and Humans

We evaluate the explainability of verbalizers using the SST-2 dataset, focusing on their consistency with human perception. To facilitate this evaluation, we utilize a manually curated list of sentiment words sourced from Hu and Liu (Hu and Liu, 2004). This curated list serves as a benchmark to assess

PLM	Method	TENWS	IFLYTEK	CLUEWSC2020	AFQMC	CSL	OCNLI	CMNLI	Avg.
BERT-base	FT	56.54	60.21	63.47	73.67	80.43	72.28	79.67	69.47
	F&P.$_{(10)}$	56.52	60.24	63.54	73.70	80.45	72.37	79.74	69.51
	F&AP.$_{(10)}$	**57.67**	**61.00**	**64.19**	**74.12**	**80.75**	**73.22**	**80.46**	**70.20** (+0.73)
BERT-wwm -ext-base	FT	56.81	59.33	62.50	74.00	80.65	74.41	80.38	69.73
	F&P.$_{(10)}$	56.90	59.28	62.49	74.02	80.61	74.31	80.39	69.71
	F&AP.$_{(10)}$	**57.43**	**59.65**	**62.99**	**74.67**	**80.86**	**75.37**	**81.04**	**70.29** (+0.56)
RoBERTa-wwm -ext-base	FT	56.88	60.30	72.13	73.97	81.07	74.66	80.44	71.35
	F&P.$_{(30)}$	56.87	60.23	72.12	73.95	81.10	74.68	80.46	71.34
	F&AP.$_{(30)}$	**57.21**	**61.26**	**72.54**	**74.19**	**81.64**	**74.87**	**80.69**	**71.77** (+0.42)
RoBERTa-wwm -ext-large	FT	58.55	62.90	81.37	76.61	82.21	78.28	82.19	74.59
	F&P.$_{(30)}$	58.45	62.87	81.43	76.57	82.18	78.22	82.19	74.56
	F&AP.$_{(30)}$	**59.28**	**63.36**	**82.22**	**77.75**	**83.16**	**78.68**	**82.93**	**75.34** (+0.75)

Table 3: Results on the development set of CLUE benchmark. Accuracy (%) is the metric used for all tasks. The number in the bracket indict the number of words selected for each class for the verbalizer. , e.g., $_{[10]}$ means select top-10 words from each class. The number in red represents the improvements of F&AP over the fine-tuning.

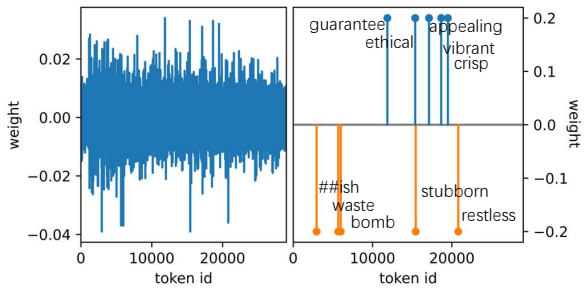

Figure 2: Denoise: left part shows all word weights, and the right shows the word weight after selection.

how well the verbalizers' vocabulary aligns with human sentiment understanding.

For the selection of verbalizers, we employ the F&P method, which identifies the top 50 words for each sentiment class based on their relevance to the dataset. The evaluation metric, depicted in the left part of Figure 3, measures the overlap between the manually curated word list and the verbalizer's selection, quantifying this as the "hit number."

Our findings indicate varying levels of consistency across different PLMs. Notably, GPT demonstrates higher consistency compared to RoBERTa-base. For instance, RoBERTa-base incorrectly categorizes certain words like *"addicted"* and *"odd"* as positive sentiments. This discrepancy partly stems from how these models' tokenizers segment words into smaller units (e.g., *"crazily"* segmented into [*"c, ##raz, ##ily"*]), which may not align with the intact sentiment words in the manual list, thus reducing the hit number.

To address the challenges posed by tokenizers, we undertook a detailed performance evaluation comparing the effectiveness of the original verbalizer against a manually curated alternative. Our approach involved meticulously identifying words shared between a manually compiled list and the vocabulary of the PLM. Subsequently, we employed a ranking methodology, selecting the top 50 words based on their classifier weights to establish the most suitable verbalizer. The outcomes of this evaluation are visually depicted in the right-hand section of Figure 3.

A smaller decrease in performance metrics indicates a closer alignment between the original and manually crafted verbalizers. Notably, the slight reduction observed in the performance of the GPT underscores the model's ability to maintain consistency and coherence with the verbalizer. This finding suggests that the verbalizer employed by GPT is inherently more transparent and interpretable, despite the challenges posed by tokenization processes.

4.3.2 Chinese Case Study

Table 4 provides real questions sampled from the OCNLI dataset. The task in OCNLI requires determining whether two given sentences are similar, which is a binary classification task. We manually constructed a template that includes two sentences for evaluation, a task description, and a [MASK] symbol. Models are tasked with predicting logits at the [MASK] position. Ideally, the token corresponding to the highest logit value should be 'yes' or another positively oriented word.

The bottom part of Table 4 displays a verbalizer obtained using the F&P method. We showcase 6 to-

Case Study
Input Demonstration:
句子 1: 一月份跟二月份肯定有一个月份有。
Sentence 1: One of January or February definitely has.
句子 2: 肯定有一个月份有。
Sentence 2: There must be a month has.
问题: 他们语义上相似吗?
Question: Are they semantically similar?
答案: [MASK]
Answer: [MASK]
Verbalizer:
1: 是，像，怡，##贴，忠，净
1: yes, like, joy, ##paste, loyal, clean
0: 变，败，##糙，罢，讳，##难
0: change, defeat, ##rough, cease, taboo, ##difficult

Table 4: Case study with a Chinese case. The upper part is a manual prompt provided to the model with its English translation. The [MASK] position in this prompt is reserved for the model to predict a logit. The lower part shows a verbalizer obtained using the F&P method, where 1 and 0 represent the positive and negative classes, respectively

kens for each class. Here, *1* represents the positive class, indicating similarity between two sentences, while *0* represents the negative class, indicating dissimilarity. It can be observed that the words in each class of the verbalizer generally correspond to the polarity expressed by that class. For instance, the list of words representing the positive class includes *yes, like, joy, ##paste, loyal, clean*. Although these tokens are not appropriate as answers to the question '*Are they semantically similar?*', their polarity aligns with human understanding.

4.4 Explain the PLM

Probing is an explainable task to detect the extent of encoded knowledge in the PLM. Linear probing (LP) (Conneau et al., 2018) is a method that only fine-tunes the linear classifier on top of the PLM on the downstream task. The predictive accuracy is interpreted as the volume of the task-related knowledge encoded in the PLM. However, during the fine-tuning, linear classifiers also encode knowledge, resulting in an overestimation in the probing results (Cao et al., 2021; Zhang and Bowman, 2018; Hewitt and Manning, 2019; Lasri et al., 2022).

As F&P does not include extra parameters, it prevents learning from fine-tuning. We freeze the PLM and only tune the linear classifier on top of the PLM. Then we construct the verbalizer according to the classifier's weight. This variant method

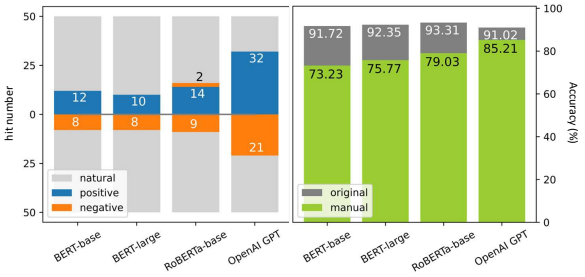

Figure 3: The left part shows how many words are both in the manual list also selected by verbalizers, i.e., hit number. The right part compares the performance of PLMs with the original verbalizer (grey) and the verbalizer constructed by the manual list (green).

is called **P**robing after **P**rompting (P&P). We conducted an experiment using the SST-2 dataset. We used AUTOPROMPT as both a baseline (Auto) and an enhancement to our approach (P&AP). The comparison results are presented in Tabel 5. The results indicate that P&P surpasses linear probing on all PLMs. This demonstrates that our method not only prevents interference from the tuning, but also maximizes the PLM's inherent potential. Furthermore, when enriched with prompts generated by AUTOPROMPT, P&AP achieved an average improvement of 7.92% over the linear probing method. The results show that combining prompting with probing is a more effective way to stimulate the most potential of PLMs.

Model	LP.	Auto.	P&P.	P&AP.
BERT-base	82.47	80.87	85.39	91.65 (+9.18)
BERT-large	84.97	82.75	86.59	91.24 (+6.27)
RoBERTa-base	85.27	91.33	86.87	92.61 (+7.34)
OpenAI-GPT	83.85	87.21	88.78	92.73 (+8.88)
Avg.	84.14	85.54	86.91	92.06 (+7.92)

Table 5: The probing result on SST-2 dataset. The number in red shows the improvements of F&AP over LP. We select the top 100 words from each class for the verbalizer in this experiment.

5 Conclusions

In this work, we propose an effective approach, referred to as F&P, which integrates fine-tuning and prompting to adapt PLMs for classification tasks. Our experimental results demonstrate that F&P yields performance comparable to fine-tuning, by employing prompts and verbalizers to replace the conventional classifier. Importantly, these prompts and verbalizers consist of real words that are easily understandable by humans. Additionally, we propose a method for assessing the explainability of

verbalizers and a variation for probing tasks. We believe that F&P not only enhances classification performance but also plays a pivotal role in demystifying the inner workings of these models.

Limitations

We summarize the limitations in two points.

Despite the significant improvement in explainability compared to traditional fine-tuning methods, F&P does not show a significant improvement in performance. This observation is frustrating because while it is important to understand and explain the decisions made by PLMs, ultimately, the performance and accuracy of these models are crucial for practical applications.

In this work, we did not discuss the effectiveness of F&P on large language models (LLMs), though LLMs are currently a prominent trend in the field. Exploring the effects of F&P on LLMs would not only provide valuable insights into the potential benefits and drawbacks of using F&P in this context but also guide future research and development in a direction that aligns with the current trends and demands of the industry.

Acknowledgments

This research work is partially supported by CUHK direct grant No. 4055209, CUHK Knowledge Transfer Project Fund No. KPF23GWP20, and the Fundamental Research Fund for the Central Universities No. CCNU24XJ004.

References

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains. *Transactions of the Association for Computational Linguistics*, 10:414–433.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Steven Cao, Victor Sanh, and Alexander Rush. 2021. Low-complexity probing via finding subnetworks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–966, Online. Association for Computational Linguistics.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2778–2788, New York, NY, USA. Association for Computing Machinery.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

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

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. OpenPrompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland. 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.

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*.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.

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.

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

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. *arXiv preprint arXiv:2204.08831*.

Chengxi Li, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, Qi Zheng, Ningyu Zhang, Yongpan Wang, and Zhi Yu. 2021. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. *ArXiv*, abs/2109.08306.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2022. Probing via prompting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1144–1157, Seattle, United States. Association for Computational Linguistics.

Jiazheng Li, Runcong Zhao, Yongxin Yang, Yulan He, and Lin Gui. 2023. Overprompt: Enhancing chatgpt through efficient in-context learning. *Preprint*, arXiv:2305.14973.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

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*.

Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com.

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.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212.

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

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Timo Schick and Hinrich Schütze. 2021. 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.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Zezhong Wang, Luyao Ye, Hongru Wang, Wai-Chung Kwan, David Ho, and Kam-Fai Wong. 2023. ReadPrompt: A readable prompting method for reliable knowledge probing. In *Findings of the Association*

for Computational Linguistics: EMNLP 2023, pages 7468–7479, Singapore. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Tianyu Liu, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022. Hpt: Hierarchy-aware prompt tuning for hierarchical text classification. *arXiv preprint arXiv:2204.13413*.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kelly W Zhang and Samuel R Bowman. 2018. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. *arXiv preprint arXiv:1809.10040*.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *Preprint*, arXiv:2108.13161.

Runcong Zhao, Qinglin Zhu, Hainiu Xu, Jiazheng Li, Yuxiang Zhou, Yulan He, and Lin Gui. 2024. Large language models fall short: Understanding complex relationships in detective narratives. *Preprint*, arXiv:2402.11051.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.