Zero-shot Approach to Overcome Perturbation Sensitivity of Prompts

Mohna Chakraborty[*], **Adithya Kulkarni**[*], and **Qi Li**
Department of Computer Science, Iowa State University
{mohnac, aditkulk, qli}@iastate.edu

Abstract

Recent studies have demonstrated that natural-language prompts can help to leverage the knowledge learned by pre-trained language models for the binary sentence-level sentiment classification task. Specifically, these methods utilize few-shot learning settings to fine-tune the sentiment classification model using manual or automatically generated prompts. However, the performance of these methods is sensitive to the perturbations of the utilized prompts. Furthermore, these methods depend on a few labeled instances for automatic prompt generation and prompt ranking. This study aims to find high-quality prompts for the given task in a zero-shot setting. Given a base prompt, our proposed approach automatically generates multiple prompts similar to the base prompt employing positional, reasoning, and paraphrasing techniques and then ranks the prompts using a novel metric. We empirically demonstrate that the top-ranked prompts are high-quality and significantly outperform the base prompt and the prompts generated using few-shot learning for the binary sentence-level sentiment classification task.

1 Introduction

The recent advance of large language models such as ChatGPT (ChatGPT, 2022), GPT-3 (Brown et al., 2020), and T5 (Raffel et al., 2020) has shown an astounding ability to understand natural languages. These pre-trained models can conduct various Natural Language Processing (NLP) tasks under the zero/few-shot settings using natural language instructions (i.e., prompts) when no or a few training samples exist. The prompts play crucial roles in these scenarios.

The prompts can be generated manually or automatically (Schick and Schütze, 2021; Gao et al., 2021; Gu et al., 2022; Wang et al., 2022). The manual prompts are handcrafted based on the user's intuition of the task (Schick and Schütze, 2021; Gao et al., 2021). Humans can easily write prompts, but the manual prompts are likely to be suboptimal since the language models may understand the instruction differently from humans. Prior studies have also shown that the performance of the language models is sensitive to the choice of prompts. For example, (Gao et al., 2021; Jiang et al., 2020) have shown that the performance is sensitive to the choice of certain words in the prompts and the position of the prompts. Due to the sensitivity and the potential misunderstanding of the instruction, manual prompts tend to suffer from poor performance under zero-shot settings. The language models tend to understand human intentions better when used with a small amount of training data. Therefore, the model can improve significantly under few-shot settings.

To address the problems of manual prompts, some studies (Jiang et al., 2020; Gao et al., 2021) further propose to generate prompts automatically following few-shot settings. These models utilize generative language models, such as the T5 model, to write automatic prompts using small training data from the task. Some studies (Shin et al., 2020) also use the small training set to fine-tune the language models or to evaluate the prompts. However, there are several drawbacks to automatically generated prompts in real applications. First, prompts cannot be generated in zero-shot settings, and the generated prompts may not follow the human intuition of the tasks. Second, deploying the generative language models also poses challenges. It can be costly to deploy on local hardware due to the size of the pre-trained generative language models. Using the generative language models via API (ChatGPT, 2022) also faces limitations, such as privacy concerns when uploading confidential customer or organizational data.

[*]equal contribution

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5698–5711

In this work[1], we aim to study how to improve manual prompts for classification tasks under zero-shot settings using moderately sized masked language models. Specifically, we use the binary sentence-level sentiment classification tasks as the testbed. Instead of deploying large generative language models, we study the usability of moderately sized masked language models, such as BERT (Devlin et al., 2019), which can be deployed and tuned in-house easily for real-world applications. The prompt follows the cloze-style format, where the position of the label is masked (e.g., "Battery life was great. The sentence was [MASK]", where a positive polarity is the goal of prediction). The prompts are used to predict probability scores for the polarity labels from the pre-trained masked language model.

To overcome the sensitivity of the language model to a manual prompt, we propose augmentation strategies to automatically generate more candidate prompts similar to the manual prompt (i.e., the base prompt), which is not required to be complex or optimized. Three augmentation techniques are designed: positioning, subordination, and paraphrasing. Different from Gao et al. (2021), where generative language models are used to generate candidate prompts, we use the same masked language models to paraphrase the base prompt. To find high-quality prompts under the zero-shot setting, we propose a novel ranking metric designed based on the intuition that high-quality prompts should be more sensitive to changing certain keywords. If a prompt is not sensitive to the change of certain keywords, it is not high-quality, and vice versa.

We conduct extensive experiments on various benchmark datasets from different domains of binary sentence-level sentiment classification and show the efficacy of the proposed ZS-SC model compared with different prompts, including manually and automatically generated prompts, in the zero-shot setting. The experimental results demonstrate the effectiveness of the proposed method in real applications.

In summary, the main contributions of this paper are as follows:

- We propose a prompt augmentation method using moderately sized masked language models to improve manual prompts for classification tasks under zero-shot settings.
- To rank the automatically generated prompts under the zero-shot setting, we propose a novel ranking metric based on the intuition that high-quality prompts should be sensitive to the change of certain keywords in the given sentence.
- Extensive experiments and ablation studies performed on benchmark datasets for sentence-level sentiment classification tasks validate the effectiveness of the proposed method.

[1]The code can be found at https://github.com/Mohna0310/ZSSC

2 Related Work

Prompt-based learning is a recent paradigm used in the zero/few-shot setting. In the zero-shot setting, the model is given a natural language instruction (prompt) describing the task without any training data (Brown et al., 2020), whereas in the few-shot setting, a few samples of training data are used along with the prompt. In prompt-based learning, the downstream tasks are formalized as masked language modeling problems using natural language prompts. Then, a verbalizer is used to map the masked language model prediction to the labels of the downstream task. This work uses prompt-based learning for the binary sentence-level sentiment classification task. This section discusses the related work that explored prompt-based learning from generic and task-specific perspectives.

Prompt-based Learning: With the introduction of GPT-3 (Brown et al., 2020), recent years have witnessed a series of studies based on prompt-based learning. Schick and Schütze (2021) utilized manual-designed hard prompts, composed of discrete words, to fine-tune the pre-trained language model. Finding the best-performing manual prompt is challenging, and to alleviate the problem, Jiang et al. (2020); Gao et al. (2021); Shin et al. (2020) designed methods for automatic prompt generation. Specifically, Shin et al. (2020) performed the downstream tasks using gradient-guided search utilizing a large number of annotations for an automatic prompt generation. Gao et al. (2021) proposed LM-BFF that auto-generates prompts using the T5 model but relies on few annotations for an automatic prompt

generation. However, the auto-generated prompts are hard prompts making them sub-optimal.

To overcome the limitations of hard prompts, Zhong et al. (2021b); Li and Liang (2021); Wang et al. (2021) proposed methods to learn soft prompts under the few-shot settings. Soft (or continuous) prompts are composed of several continuous learnable embeddings, unlike hard prompts. Motivated by the prior studies, Zhao and Schütze (2021) utilized both the hard and soft prompts for training the pre-trained language model. Gu et al. (2022) proposed pre-training hard prompts by adding soft prompts into the pre-training stage to obtain a better initialization.

Another line of study (Khashabi et al., 2022; Wang et al., 2022; Zhong et al., 2021a) designed manual task-specific prompts by fine-tuning pre-trained language models on multiple tasks. The fine-tuned language model is then used on unseen tasks under the zero/few-shot setting.

Prompt-based Learning for Sentence-level Sentiment Classification: Over the past years, a large body of studies (Shin et al., 2020; Gao et al., 2021; Gu et al., 2022; Wang et al., 2022) have demonstrated excellent performance in few-shot settings on sentence-level sentiment classification tasks. Specifically, Shin et al. (2020) used gradient-guided search to generate automatic prompts, whereas Gao et al. (2021) used a more general-purpose search method to generate automatic prompts. Following the limitation of automatic prompts, Gu et al. (2022) suggested hybrid training combining hard and soft prompts in the initial stage, obtaining a better initialization. Wang et al. (2022) proposed a Unified Prompt Tuning framework and designed prompts by fine-tuning a pre-trained language model over a series of non-target NLP tasks and using the trained model to fit unseen tasks. For instance, when the target task is sentiment classification, the training data is from other domains like NLI and paraphrasing.

These studies consider access to labeled instances and perform the sentence-level sentiment classification task using a large-scale pre-trained generative language model. In our study, we do not use any training data, and the base prompt can be considered as a natural language description for the task. Therefore, this study follows the zero-shot setting. Using a moderately sized masked language model further makes the proposed method more appealing in practice.

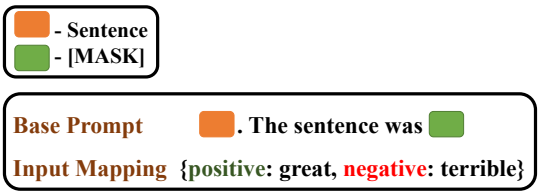

Figure 1: Model Input

3 Methodology

This section first discusses the problem formulation and the overview in Section 3.1 and Section 3.2. Our proposed method handles the language model's sensitivity to a manual prompt by utilizing prompt augmentation techniques to generate multiple candidate prompts. The detailed description of the prompt augmentation is discussed in Section 3.3. To rank the automatically generated prompts in the zero-shot setting, we propose a novel ranking metric, discussed in Section 3.4. Finally, the top-ranked prompts are used for prediction, discussed in Section 3.5.

3.1 Problem Formulation

Given an unlabeled corpus $\mathcal{D}$ with N sentences, an input mapping $\mathcal{M} : \mathcal{Y} \rightarrow \mathcal{V}$ for the labels $y \in \mathcal{Y} = \{-1, 1\}$, in the vocabulary $\mathcal{V}$ of $\mathcal{L}$ and a base prompt B_p, the task is to find quality prompts similar to the base prompt in a zero-shot setting for the binary sentence-level sentiment classification task. Figure 1 shows one example input to the model. In this example, $y \in \mathcal{Y} = \{negative, positive\}$, $\mathcal{M}(positive) = great$, and $\mathcal{M}(negative) = terrible$.

3.2 Overview

Given a base prompt B_p, the proposed ZS-SC first generates multiple prompts similar to the base prompt using augmentation techniques. Specifically, we introduce positioning, subordination, and paraphrasing techniques in the augmentation process, which are discussed in detail in Section 3.3.

With more automatically generated candidate prompts, ZS-SC ranks the prompts using a novel ranking metric. This metric is designed based on the observation that quality prompts should flip the predicted label if $\mathcal{M}(y)$ present in the sentence is replaced with $\mathcal{M}(y')$, where $y \neq y'$, whereas the predicted label should stay the same if $\mathcal{M}(y)$ is replaced with its synonyms. Section 3.4 discusses the proposed ranking metric in detail.

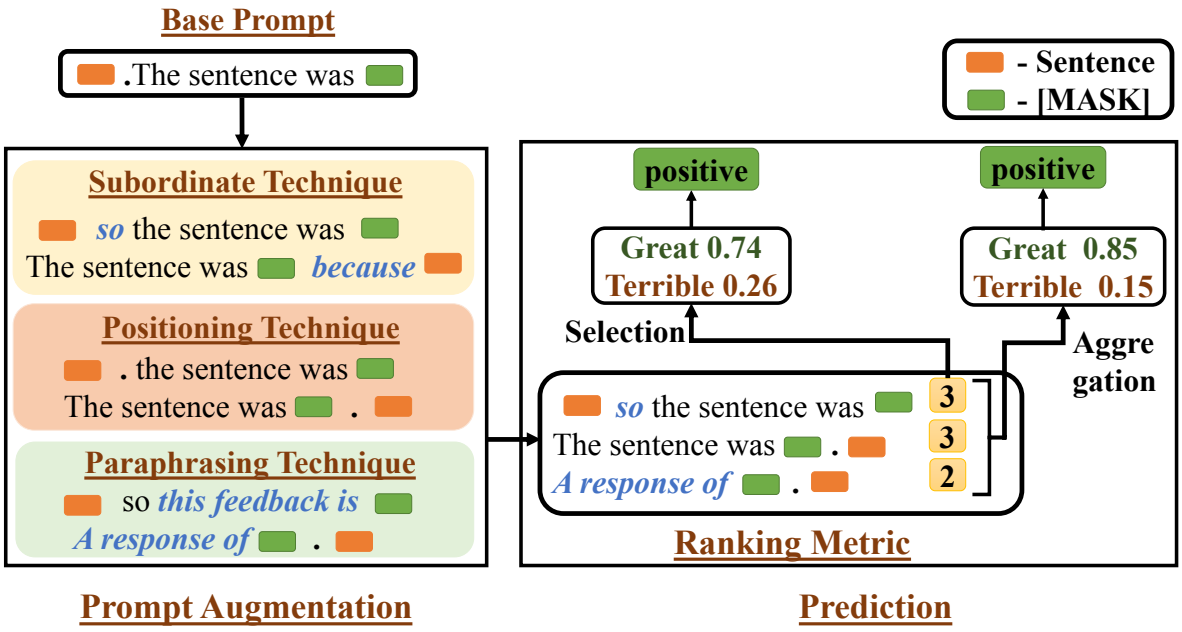

Figure 2: Overview of ZS-SC.

Finally, the top-ranked prompt is selected, or top$-k$ highly ranked prompts are aggregated to conduct the zero-shot prediction for the unlabeled corpus $\mathcal{D}$ (Section 3.5).

Figure 2 illustrates the overview of the proposed approach, ZS-SC.

3.3 Prompt Augmentation

A single base prompt provided by a user may not provide optimal results for the given task. Prior studies (Gao et al., 2021; Jiang et al., 2020) have shown that the performance of the prompts is sensitive to the choice of certain words and the position of the prompts, respectively. Furthermore, we observe that using subordinate conjunctions to join the prompt and sentence can improve the method's performance on some datasets since it introduces a dependency between the prompt and sentence, thereby leading the model to relate the predicted label with the context of the sentence. Based on the above observations, we propose to apply three augmentation techniques to generate prompts automatically, namely positioning, subordination, and paraphrasing techniques.

The *positioning* technique places the prompt either before or after the given sentence. The *subordination* technique uses subordinate conjunctions like *"because" and "so"* to join the prompt and the sentence. Specifically, the conjunction *"because"* is used if the prompt is placed before the sentence, and the conjunction *"so"* is used if the prompt is placed after the sentence.

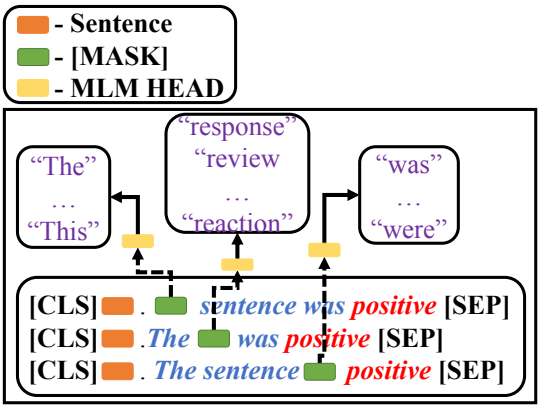

Figure 3: Paraphrasing Technique.

The *paraphrasing* technique generates multiple prompts similar to the base prompt B_p by swapping the tokens in the base prompt with similar tokens. These similar tokens should have the same part of speech tags as the tokens they are replacing and should not change the context of the prompt. Therefore, to obtain these similar tokens, we use a pre-trained MLM model $\mathcal{L}$. Pre-trained MLM models are trained to predict the missing tokens that fit the context of the given sentence and thus would be suitable for the purpose. Figure 3 illustrates the paraphrasing technique for the base prompt. The label "positive" is used as a

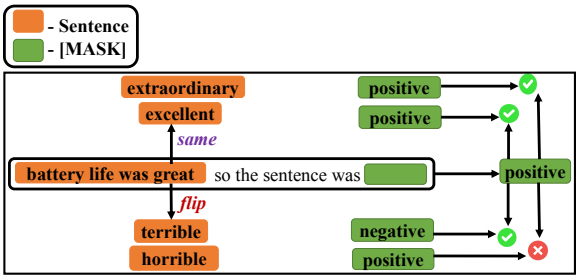

Figure 4: Ranking Metric.

placeholder so that pre-trained MLM model can learn the context of the given sentence.

If a specific sentence is joined with the base prompt, the MLM model $\mathcal{L}$ can understand the context better, so the replacing tokens will make more sense. Therefore, instead of using prompts alone, we form sample instances by randomly selecting sentences from the unlabeled corpus $\mathcal{D}$. We then mask the replaceable tokens from the base prompt one at a time and use the MLM model $\mathcal{L}$ to predict the masked token. For each masked token, the MLM model $\mathcal{L}$ gives a score to all the tokens in its vocabulary. We choose the top-K ranked tokens as similar token candidates and remove those that do not have the same POS tag as the masked token.

These three techniques can be applied in different combinations and permutations to generate prompts automatically. The number of candidate paraphrasing tokens K can be increased to generate more prompts. Figure 3 illustrates the process of obtaining paraphrasing tokens to the tokens of the base prompt.

3.4 Ranking Metric

Not all the automatically generated prompts in Section 3.3 obtain good performance for the task. Therefore, we aim to rank these prompts and choose quality prompts for the tasks. Previous works (Gao et al., 2021; Shin et al., 2020) have used validation or manually annotated few-shot training data for evaluating the automatically generated prompts. However, under the zero-shot setting, we do not assume there exists any manually annotated data. Therefore, we have to rank the automatically generated prompts in the absence of manually annotated data which is not considered by the previous works.

Intuitively, if the mapping token of the opposite label replaces the mapping token in a given sentence, the predicted label by a quality prompt should flip. On the other hand, the predicted label should remain the same if the mapping token in the sentence is replaced by its synonyms. For example, suppose we replace the word *"great"* in sentence *"battery life was great"* with *"terrible"*. In this case, the predicted label should flip, whereas if we replace *"great"* with *"excellent"*, the predicted label should remain the same. We use this intuition to measure the sensitivity of the prompt to the change of the mapping tokens in the given sentences. The measured sensitivity implies the quality of the prompt, namely prompts sensitive to the change of the mapping tokens in the given sentence can achieve good performance for the task. Figure 4 illustrates the key idea of the proposed ranking metric.

We model the above intuition as a zero-one scoring function. To do so, we first obtain sentences from the unlabeled corpus $\mathcal{D}$ that contain the mapping tokens $\mathcal{M}(y) \in \mathcal{V}$ obtained from the provided input mapping $\mathcal{M} : \mathcal{Y} \rightarrow \mathcal{V}$. If the mapping tokens are not present in the corpus $\mathcal{D}$, the synonyms of the mapping tokens can be used.

For a sentence $s_{in} \in S_W$, let the label predicted by the model for a given prompt P be l_1. We then replace the mapping token $\mathcal{M}(y)$ in s_{in} with $\mathcal{M}(y')$, where $y \neq y'$ to obtain a new sentence s'_{in}. Let the label predicted for s'_{in} be l_2. The zero-one scoring function for this scenario is defined as:

$$\lambda_{s_{in}} = \begin{cases} 1, & \text{if } l_2 \neq l_1 \\ 0, & Otherwise \end{cases}. \tag{1}$$

We consider the synonyms of $\mathcal{M}(y)$ to further diversify the scoring function. Specifically, we use Wordnet (Miller, 1995) to obtain synonyms for $\mathcal{M}(y)$. We replace $\mathcal{M}(y)$ by its synonym to obtain a new sentence s''_{in}. Let the label predicted for s''_{in} be l_3. The scoring function for this scenario is defined as:

$$\lambda_{s_{in}} = \begin{cases} 1, & \text{if } l_3 = l_1 \\ 0, & Otherwise \end{cases}. \tag{2}$$

Similarly, we can also consider the synonyms of $\mathcal{M}(y')$. The predicted label should flip if $\mathcal{M}(y)$ is replaced by synonyms of $\mathcal{M}(y')$.

Let Z be the set of new sentences obtained through synonym replacement. The overall score for a given prompt (P) is defined as:

$$Score(P) = \sum_{i=1}^{|S_W|} \sum_{j=1}^{|Z|} \lambda_{s_{ij}}. \tag{3}$$

A higher score indicates that the prompt is more sensitive to the polarity of mapping tokens.

The score is calculated for all the prompts generated in the prompt augmentation step (Section 3.3), and then the prompts are ranked based on their calculated score. The top-ranked prompt is the prompt with the highest score. Figure 4 depicts the functioning of our ranking metric.

3.5 Prediction

First, we define how we obtain the prediction probabilities using any given prompt. Given an input mapping $\mathcal{M} : \mathcal{Y} \rightarrow \mathcal{V}$ that maps the task label space to individual words in the vocabulary $\mathcal{V}$ of pre-trained MLM model $\mathcal{L}$, the probability of a label $y \in \mathcal{Y}$ for a given sentence s_{in} in the unlabeled corpus $\mathcal{D}$ using a prompt P is obtained as:

$$p(y|s_{in}) = p([MASK] = \mathcal{M}(y)|s_P) = \frac{exp(w_{\mathcal{M}(y)}.h_{[MASK]})}{\sum_{y' \in \mathcal{Y}} exp(w_{\mathcal{M}(y')}.h_{[MASK]})}, \quad (4)$$

where $s_P = P(s_{in})$ is the sentence s_{in} joined with the prompt P, which contains exactly one masked token at the position of the label, $h_{[MASK]}$ is the hidden vector of the [MASK] token and w_v is the pre-softmax vector corresponding to $v \in \mathcal{V}$. The predicted label for the given sentence s_{in} is the label y with the highest probability.

Our proposed approach is to use quality prompts for the zero-shot prediction tasks. We can either select the top-ranked prompt or aggregate top-k-ranked prompts. If the top-1 prompt is selected, Eq. (4) is used to obtain the label probability for each sentence, and the label with the highest probability is the predicted label.

Prompt aggregation may help correct the mistakes of the individual prompts. We consider prediction confidence and use the soft labels computed by Eq. (4) in aggregation. Let $p_1(y), p_2(y), .., p_k(y)$ be the prediction probability for label $y \in \mathcal{Y}$ obtained using top-k prompts. The aggregated prediction probability is:

$$p(y) = \frac{\sum_{i=1}^{k} Score(p_i) * p_i(y)}{\sum_{i=1}^{k} Score(p_i)}, \quad (5)$$

and then the label with the highest aggregated prediction probability is chosen for the sentence.

Table 1: Statistics of the Datasets

Datasets	SST-2		MR		CR	
	Pos	**Neg**	**Pos**	**Neg**	**Pos**	**Neg**
Train	3610	3310	4331	4331	1407	368
Dev	444	428	0	0	0	0
Test	909	912	1000	1000	1000	1000
Total	4963	4650	5331	5331	2407	1368

4 Experiments

In this section, we evaluate the proposed ZS-SC model on several benchmark binary sentence-level sentiment classification datasets from various domains. More studies can be found in the Appendix A.

4.1 Dataset

The performance of ZS-SC is evaluated across 3 widely used sentiment classification datasets: SST-2 (Socher et al., 2013), MR (PANG, 2002), and CR (Hu and Liu, 2004). The dataset statistics are provided in Table 1.

4.2 Evaluation Metrics

Since no training data is used in zero-shot settings, we evaluate all prompts on the *entire dataset*. We use **Accuracy (Acc.)** and **macro F1 score (F1)** for all the datasets to evaluate the performance of ZS-SC and compare it with baselines under different settings. Note that Accuracy is equivalent to micro F1 score in binary classification tasks.

4.3 Baseline Methods

Since none of the prior work has performed the task of binary sentence-level sentiment classification under the zero-shot setting, we compare it with the baselines that have performed the task under the few-shot setting for the datasets discussed in Section 4.1. For a fair comparison, we modified these studies as per the zero-shot setting, using the prompts reported in their paper. The baseline templates are discussed in Table 5 of Appendix A.

LM-BFF (Gao et al., 2021): This paper explores manual prompts and generates automatic prompts under the few-shot setting. Specifically, they use few-shot examples to automatically generate prompts using the T5 model. The performance of their method is evaluated on a range of classification and regression tasks using RoBERTa-large (Liu et al., 2019) with fine-tuning. We compare ZS-SC with their manual prompt and their top-ranked automatic prompts.

Table 2: Results of the sentiment classification task on the three benchmark datasets using BERT base and BERT large. We report accuracy and F1 score for all datasets. The results are evaluated on the entire dataset. We report the majority voting results for the automatic prompt baselines. The best-performing and runner-up model per column are highlighted in bold and underlined, respectively.

Method	Prompt	BERT base						BERT large					
		SST-2		MR		CR		SST-2		MR		CR	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
LM-BFF	Automatic	58.46	62.24	57.94	62.81	71.35	69.66	52.69	59.33	57.3	63.69	70.55	69.11
UPT		57.46	61.79	62.65	66.78	75.09	73.53	53.82	61.08	65.2	69.69	72.62	71.4
LM-BFF	Manual	62.3	65.75	58.18	62.16	74.9	72.81	61.15	65.41	57.88	62.64	72.59	70.85
PPT		52.53	56.93	50.5	53.41	64.03	61.02	52.29	57.68	50.5	56.0	63.9	62.21
Base Prompt†		62.3	65.75	58.18	62.16	74.9	72.81	61.15	65.41	57.88	62.64	72.59	70.85
Base Prompt⋆		63.22	63.15	59.97	60.25	69.04	64.29	54.12	58.6	54.43	57.12	56.59	62.14
ZS-SC (Top-1)†	Automatic	67.48	67.52	58.93	62.07	73.36	70.16	74.13	75.66	69.84	71.75	73.12	70.65
ZS-SC (Top-3)†		67.12	68.22	60.15	60.14	71.19	68.23	67.58	70.65	64.15	67.91	70.05	67.82
ZS-SC (Top-5)†		67.99	68.94	61.19	62.92	71.51	69.32	66.55	70.09	63.47	67.76	69.41	67.32
ZS-SC (Top-1)⋆		**72.18**	**72.36**	**68.24**	**68.26**	75.09	72.1	74.74	74.71	70.29	70.36	80.47	78.43
ZS-SC (Top-3)⋆		71.92	72.01	67.88	67.89	76.82	74.43	**77.11**	**77.58**	**72.96**	**73.54**	79.17	77.84
ZS-SC (Top-5)⋆		71.5	71.46	66.74	66.88	**77.26**	**74.52**	76.9	77.54	72.46	73.43	**81.45**	**79.52**

PPT (Gu et al., 2022): This paper proposes pre-training hard prompts by adding soft prompts to achieve better initialization into the pre-training stage on classification tasks. ZS-SC is compared with their manual prompt.

UPT (Wang et al., 2022): This paper proposes a Unified Prompt Tuning framework and designs prompts by fine-tuning a pre-trained language model (RoBERTa-large) over a series of non-target NLP tasks. After multi-task training, the trained model can be fine-tuned to fit unseen tasks. ZS-SC is compared with their top-ranked prompts.

4.4 Settings

The experiments are conducted using pre-trained uncased BERT (BERT base and BERT large) encoders. BERT base has 12 attention heads, 12 hidden layers, and a hidden size of 768 resulting in 110M pre-trained parameters, whereas BERT large has 16 attention heads, 24 hidden layers, and a hidden size of 1024 resulting in 336M pre-trained parameters. We set K, the hyperparameter for the number of candidate words in paraphrasing, to 30. We obtain 6 synonyms for each mapping word from WordNet (Miller, 1995). The size of the set of new sentences through synonym replacement (Z) is 12, 6 of which are obtained by replacing the mapping token $\mathcal{M}(y)$ with its synonyms, and the other 6 are obtained by replacing the mapping token by $\mathcal{M}(y')$ and synonyms of $\mathcal{M}(y')$, where $y \neq y'$.

For ZS-SC, we considered two different base prompts. The first base prompt is *"<sentence>. It was [MASK]"*, which is the same as the manual prompt used by LM-BFF (denoted by † in Table 2), whereas the second base prompt is *"<sentence>. The sentence was [MASK]"* (denoted by ⋆ in Table 2). The base prompts defined are generic and used for all datasets.

4.5 Results and Discussion

To better compare the performance of different methods, we categorize them based on the prompt (manual or automatic).

Table 2 shows the results of all prompts using BERT base and BERT large pre-trained MLM models, respectively. ZS-SC with the ⋆ base prompt significantly outperforms both manual and automatic baseline methods on both pre-trained MLM models on all three datasets. Overall, the aggregation strategy tends to outperform the selection strategy, but the outperformance is inconsistent across different data. We conduct more studies on the impact of top-k prompts in Section 4.6.

It is interesting to notice that for † base prompt ZS-SC outperforms on SST-2 and MR datasets but not on the CR dataset. Furthermore, the margin of ZS-SC over the base prompt decreases for † compared to ⋆ base prompt. This is because "It was" is harder to augment than "The sentence was" since the former is shorter and contains no concrete word. Even though the † base prompt is not ranked top-1 by ZS-SC on the CR dataset, it is ranked as the 4-th for both pre-trained MLM models, demonstrating that ZS-SC can recognize † base prompt as a high-quality prompt.

It is also interesting to note that for baseline methods, either using manual or automatic prompts, there is no significant gain using the BERT large over the BERT base encoder, and the performance

Table 3: Ablation study results with and without WordNet on the three benchmark datasets for sentiment classification tasks. We report accuracy and F1 score for all datasets using BERT base and BERT large. The results are evaluated on the entire dataset.

Method	Encoder	SST-2		MR		CR	
		Acc.	F1	Acc.	F1	Acc.	F1
ZS-SC-W (Top-1)	BERT base	62.77	64.14	59.25	63.3	72.04	71.29
ZS-SC-W (Top-3)		62.57	65.73	60.1	64.34	75.78	72.76
ZS-SC-W (Top-5)		62.85	66.41	61.0	64.91	75.67	73.63
ZS-SC (Top-1)		72.18	72.36	68.24	68.26	75.09	72.1
ZS-SC (Top-3)		71.92	72.01	67.88	67.89	76.82	74.43
ZS-SC (Top-5)		71.5	71.46	66.74	66.88	77.26	74.52
ZS-SC-W (Top-1)	BERT large	73.55	74.1	70.29	70.36	80.47	78.43
ZS-SC-W (Top-3)		74.54	75.0	69.94	71.03	79.17	77.83
ZS-SC-W (Top-5)		75.68	76.74	71.89	73.14	81.0	78.94
ZS-SC (Top-1)		74.74	74.71	70.29	70.36	80.47	78.43
ZS-SC (Top-3)		77.11	77.58	72.96	73.54	79.17	77.84
ZS-SC (Top-5)		76.9	77.54	72.46	73.43	81.45	79.52

of a prompt can change significantly using different pre-trained language models. However, we can observe that the performance of ZS-SC improves with the scale of the model. The key difference between ZS-SC and the automatic prompts generated by baseline models is that we use the same language models to generate prompts and conduct classification tasks, whereas baselines generate prompts manually or using a different model. These results suggest that different language models have different knowledge of the language, so prompts need to be generated specifically for the chosen language model.

4.6 Study of Selection VS Aggregation

Comparing top-1 selection to top-k aggregation, from Table 2, we can observe that top-1 selection performs better compared to top-k aggregation on BERT base whereas on BERT large top-k aggregation performs better. Furthermore, we can observe that the top-k aggregation result does not increase with k as suggested by previous works (Gao et al., 2021).

To further analyze our observation, we plot the change in performance of ZS-SC with respect to the number of aggregated top-k prompts for BERT large encoder on $\star$ base prompt in Figure 5. Figure 5 shows that the top-k aggregation performance increases with k only for SST-2 dataset and does not increase for CR and MR datasets. This implies that top-k aggregation performance increases with k only for some datasets but not all. Furthermore, we can also observe that top-k aggregation performance can be better than top-1 selection performance on all three datasets. We believe that aggregation performance improves when the top-ranked prompts make independent mistakes.

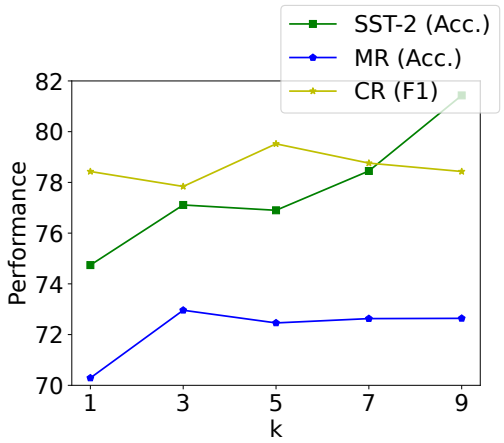

Figure 5: Performance vs the number of aggregated top-k prompts for BERT large on $\star$ base prompt.

4.7 Study of the Proposed Ranking Metric

To study the effectiveness of the proposed ranking metric, we plot the accuracy of the augmented prompts evaluated using ground truth labels with respect to their ranks based on the proposed ranking metric. The results for SST-2 dataset using the BERT base model on $\star$ base prompt are shown in Figure 6. The figure shows that the highly-ranked prompts achieve higher accuracy than the low-ranked prompts in general, demonstrating the effectiveness of our proposed ranking metric. Furthermore, we can observe that the accuracy of the prompts decreases as the rank provided by our proposed ranking metric increases.

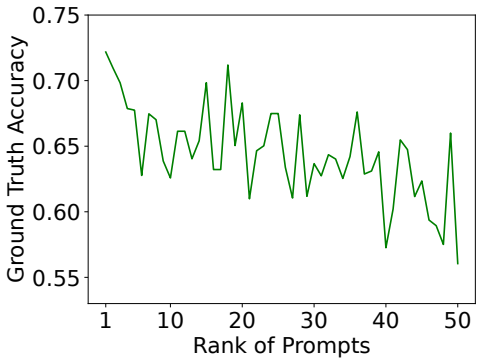

Figure 6: Proposed Metric vs Ground Truth Performance for SST-2 dataset using BERT base model on ⋆ base prompt .

4.8 Ablation Studies

We conduct ablation studies to investigate the contributions of Wordnet synonyms to the overall model performances.

Table 3 shows the performance of ZS-SC with and without Wordnet. From the results, we can observe that ZS-SC with Wordnet outperforms ZS-SC without Wordnet for both variants of pre-trained MLM models. The results show that diversification of the mapping tokens helps the scoring function to rank the prompts better and subsequently improve the performance.

5 Conclusion

This work proposes to study how to improve manual prompts for binary sentence-level sentiment classification tasks under zero-shot settings. To overcome the sensitivity of the language model to a manual prompt, we propose prompt augmentation techniques to generate multiple candidate prompts. Further, to rank the generated prompts without labeled data, we propose a novel ranking metric based on the intuition that high-quality prompts should be sensitive to the change of certain keywords in the given sentence. Extensive experiments and ablation studies demonstrate the power of the proposed ZS-SC on three benchmark datasets.

Limitations

The proposed method is tested for a binary labeling scenario where each instance can belong to one of the labels but not both. The scenario of overlapping labeling space is not tested, nor is the scenario for multi-class labeling space. Since we aim to obtain high-quality prompts similar to the base prompt, if the base prompt is very restrictive, then the suggested prompt might be the same as the base prompt. The approach only applies to two moderately sized MLM models, and the extension to other larger models is not tested.

Ethics Statement

We comply with the ACL Code of Ethics.

References

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.

ChatGPT. 2022. Chatgpt: Optimizing language models for dialogue. In *OpenAI. Retrieved from https://openai.com/blog/chatgpt/, Access Date: 16.12.2022.*

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.

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.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. Ppt: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423.

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.

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.

Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to gptk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

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*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

B PANG. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

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.

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*, pages 1631–1642.

Chengyu Wang, Jianing Wang, Minghui Qiu, Jun Huang, and Ming Gao. 2021. Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2792–2802.

Jianing Wang, Chengyu Wang, Fuli Luo, Chuanqi Tan, Minghui Qiu, Fei Yang, Qiuhui Shi, Songfang Huang, and Ming Gao. 2022. Towards unified prompt tuning for few-shot text classification. *arXiv preprint arXiv:2205.05313*.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021a. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021b. 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.

A Appendix

A.1 Case Study

Table 4 shows the top-5 ranked prompts for three datasets, SST-2, MR, and CR. The table shows that prompts with subordinate conjunctions like *"because"* and *"so"* are ranked higher. The ranking confirms our intuition that subordinate conjunctions that introduce a dependency between the prompt and the sentence can improve the performance of the prompts. Note that the proposed ranking metric ensures that low-quality prompts are not ranked higher. Therefore the results from the table suggest that prompts with subordinate conjunctions are high-quality.

Table 4: Top 5 Ranked Prompts for BERT large and BERT base

Dataset	BERT large	BERT base
SST-2	The sentence sounded [MASK] because <sentence> . Every sentence was [MASK] . <sentence> . <sentence> . Every sentence was [MASK] . The result was [MASK] . <sentence> . Each sentence was [MASK] . <sentence> .	<sentence>. Every sentence was [MASK] . Every sentence was [MASK]. <sentence> . Each sentence was [MASK] . <sentence> . <sentence>. Each sentence was [MASK] . <sentence> so every sentence was [MASK] .
MR	The sentence sounded [MASK] because <sentence> . The sentence seemed [MASK] because <sentence> . The result was positive . <sentence> . Every sentence was [MASK] because <sentence> . Every sentence was [MASK] . <sentence> .	<sentence>. Every sentence was [MASK] . Every sentence was [MASK]. <sentence> . Each sentence was [MASK] . <sentence> . <sentence> . Each sentence was [MASK] . <sentence> so the sentence sounded [MASK] .
CR	The sentence sounded [MASK] because <sentence> . The sentence sounded [MASK] . <sentence> . <sentence> . The sentence sounded [MASK] . Every sentence was [MASK] . <sentence> . The answer was [MASK] . <sentence> .	The sentence sounded [MASK] . <sentence> . <sentence> . The sentence sounded [MASK] . Every sentence was [MASK] . <sentence> . <sentence> . Every sentence was [MASK] . This sentence was [MASK] . <sentence> .

Table 5: Ranked Prompts of Baselines

Dataset	LM-BFF	PPT	UPT
SST-2	<sentence>. A [MASK] one. <sentence>. A [MASK] piece. <sentence>. All in all [MASK].	<sentence>. [MASK].	<sentence>. It was [MASK]. <sentence>. I thought it was [MASK]. <sentence>. It is [MASK]. <sentence>. The review is [MASK]. <sentence>. A [MASK] one.
MR	It was [MASK] ! <sentence>. <sentence>. It's [MASK]. <sentence> A [MASK] piece of work.	<sentence>. [MASK].	<sentence>. A [MASK] piece of work. <sentence>. It is [MASK]. <sentence>. The film is [MASK]. <sentence>. A really [MASK] movie.
CR	<sentence>. It's [MASK] ! <sentence>. The quality is [MASK]. <sentence>. That is [MASK].	<sentence>. [MASK].	<sentence>. It was [MASK]. <sentence>. It looks [MASK]. <sentence>. It is [MASK]. <sentence>. The quality is [MASK]. <sentence>. I thought it was [MASK].

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitation

☑ A2. Did you discuss any potential risks of your work?
Limitation

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction (Section 1)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
The results are deterministic.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.