Do Prompts Solve NLP Tasks Using Natural Language?

Anonymous ACL submission

Abstract

Thanks to the advanced improvement of large pre-trained language models, prompt-based fine-tuning is shown to be effective on a variety of downstream tasks. Though many prompting methods have been investigated, it remains unknown which type of prompts are the most effective among three types of prompts (i.e., human-designed prompts, schema prompts and null prompts). In this work, we empirically compare the three types of prompts under both few-shot and fully-supervised settings. Our experimental results show that schema prompts are the most effective in general. Besides, the performance gaps tend to diminish when the scale of training data grows large.

1 Introduction

Prompt-based fine-tuning has gained increasing attention on NLP (Shin et al., 2020; Schick et al., 2020; Schick and Schütze, 2021b; Paolini et al., 2021; Gao et al., 2021). The main idea is to leverage knowledge in pre-trained language models for downstream tasks, by reformulating a specific task into the form of language modeling tasks, with the aid of prompts. Among various recent methods on prompt-based NLP, there has been three major forms of prompts, which we call *NL template prompts*, *schema prompts* and *null prompts*, respectively. NL template prompts (Petroni et al., 2019; Jiang et al., 2020; Gao et al., 2021) were the earliest proposed and the dominant method. As illustrated in Table 1, they use a natural language sentence to augment a given input, where the added prompt contains a mask token that indicates the output class. In contrast, schema prompts (Lee et al., 2021; Paolini et al., 2021) replace a natural language sentence with a structured schema, which makes the prompt more succinct and code-like. Null prompts (Logan IV et al., 2021) are the most succinct version, directly adding a masked token to the end of the input.

<Input>
"The movie fails to live up to the sum of its parts."
Template Prompt:
`[CLS]` *<Input> It was* `[MASK] [SEP].`
Schema Prompt:
`[CLS]` *<Input> Sentiment:* `[MASK] [SEP].`
Null Prompt (Logan IV et al., 2021)**:**
`[CLS]` *<Input>* `[MASK] [SEP].`

Table 1: An example of different types of prompts, from SST-2 dataset.

While different types of prompts have been compared for specific tasks (Gao et al., 2021; Logan IV et al., 2021), there has been little work systematically comparing their effects over a large variety of tasks and training settings (i.e., few shot). We aim to fill the gap by empirically addressing the following three research questions:

First, which type of prompt is generally the most effective? Intuitively, natural language prompts better connect large pre-training and task fine-tuning by having the same language style in both phases. However, it can increase the difficulty of representation by introducing overly long sequence extensions. In contrast, schema and null prompts are more succinct, but less close to natural language pre-training.

Second, are task-specific information useful to include in prompts. Compared with NL templates and schemas, null prompts are the most succinct, and are task-agnostic in not including any task hints in the augmented sequence. While having been shown effective for several NLI-style classification tasks under few-shot settings (Logan IV et al., 2021), it remains a question whether they are competitive in more general settings.

Third, what is the effect of automatically searching for prompt template and masked label words? There has been a line of work automatically finding prompts, which results in seemingly unnatural augmented sequences (Shin et al., 2020; Gao et al., 2021). In addition, the words to use for filling the masked output slots are also flexible. We want to learn whether these automatic selections have significant benefit compared with human definitions.

Results show that among the three types of prompts, schema prompts are the most effective in general. However, the gap between the three types of prompts tends to diminish when the scale of training data grows sufficiently large. Finally, both automatic templates and automatic tokens give better results compared with more understandable human prompts. Our code will be released at https://github.com/anonymous.

2 Experimental Setup

2.1 Basic Settings

We mainly experiment on sentence classification tasks, which have been extensively investigated in previous work (Schick and Schütze, 2021a; Gao et al., 2021). We also include two structure prediction tasks (i.e., NER and relation classification) for generalization beyond sentence classification. For sentence classification, we follow Gao et al. (2021) to adopt `RoBERTa-large` (Liu et al., 2019) and conduct experiments on eight sentence classification datasets For structure prediction, we use the method and setting of Cui et al. (2021), which formulized NER as a text generation task. We experiment with CoNLL03 Dataset for NER and TACRED Dataset for relation classification. We adopt the same hyperparameters used in previous work.

To get a comprehensive view, we experiment with both few-shot and rich-resource settings. We also report standard deviation in the few-shot experiments. More details about experimental settings can be found in Appendix A.1.

2.2 Prompt

We experiment with template prompts, schema prompts and *null* prompts (Logan IV et al., 2021). Following Schick and Schütze (2021a), a prompt method generally contains a *pattern* that maps inputs to prompt-style outputs and a *verbalizer* that maps labels to vocabulary tokens. In this paper, the term "prompt" normally refers to the *pattern*, while the term "label word" refers to the verbalizer. Some prompt examples are shown in Tables 2 and 3.

Dataset	*<Pattern>* Template	*<Pattern>* Schema
SST-2	It was	Sentiment:
SST-5	It was	Sentiment:
MR	It was	Sentiment:
CR	It was	Sentiment:
MPQA	It was	Opinion:
Subj	This is	Opinion:
TREC	–	Question type:
CoLA	This is	Grammatical:

Table 2: Examples of template-based and schema-based prompts of various sentence classification tasks. The prototype prompt is formulized as "`[CLS]`*<Input>* *<Pattern>* `[MASK] [SEP].` ", except that the template prompt of TREC is "`[CLS] [MASK]:` *<Input>* `[SEP].` " which does not require any patterns.

CoNLL03
Templae:
<Input>. [span] is a *person* entity.
Schema:
<Input>. [span]: *person* entity.
TACRED
Template:
<Input>. The relation between [span_1] and [span_2] is *no_relation*.
Schema:
<Input>. [[span_1] | [span_2]] relation: *no_relation*.

Table 3: Example prompts for structure prediction tasks, where [span] refers to a text span in the input sentence (i.e., *<Input>*) and the italic parts (e.g, *person* and *no_relation*) are the entity or relation types. For more details, please refer to Cui et al. (2021).

2.3 Prompt Types

Tables 2 and 3 give examples for different prompt patterns.

Template Prompt We define template prompts are fluent sentences that contains task-specific hints. Our template-based prompts for sentence classification and structure prediction are respectively from Gao et al. (2021) and Cui et al. (2021).

Schema Prompt We define schema-based prompts as syntactically-incorrect but task-related prompts. We design schema-based prompts that are unnatural to human speaking and writing, using our intuition about the specific tasks. We do not further tune any of these prompts in our experiments. Although this may introduce subjective bias to our experiments, we argue that another different set of schema-based prompts would not make significant difference when comparing with template-based prompts.

Null Prompt Logan IV et al. (2021) proposed to use *null* prompts in few-shot prompt fine-tuning:

		SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (Matt.)	# Wins
Few-shot	Fine-tuning	81.4 (3.8)	43.9 (2.0)	76.9 (5.9)	75.8 (3.2)	72.0 (3.8)	90.8 (1.8)	88.8 (2.1)	**33.9 (14.3)**	1
	Template Prompt	92.6 (0.6)	47.2 (1.3)	87.1 (1.9)	90.7 (0.9)	84.5 (2.3)	91.3 (1.2)	85.8 (2.4)	9.2 (6.7)	0
	w/ auto label word	92.4 (1.0)	43.6 (1.4)	86.3 (2.4)	90.2 (1.1)	85.8 (1.7)	91.2 (1.1)	88.7 (3.3)	13.9 (14.3)	0
	w/ special token	91.5 (1.3)	45.5 (1.3)	84.7 (1.5)	86.5 (4.7)	73.5 (6.6)	90.4 (2.4)	84.6 (2.9)	11.2 (7.9)	0
	Schema Prompt	93.2 (0.1)	**50.2 (0.7)**	87.3 (1.0)	91.6 (0.6)	85.2 (1.5)	91.4 (0.5)	87.8 (2.2)	9.6 (3.0)	1
	w/ auto label word	**93.6 (0.6)**	47.9 (1.1)	**87.5 (1.6)**	**91.7 (0.7)**	**86.0 (0.5)**	**91.9 (0.9)**	89.2 (2.1)	15.0 (2.5)	5
	w/ special token	92.0 (1.3)	48.8 (3.5)	86.4 (2.1)	87.0 (3.7)	68.5 (3.3)	90.4 (0.9)	**90.6 (1.4)**	14.6 (5.2)	1
	Null Prompt	92.7 (0.6)	49.0 (1.1)	86.4 (1.3)	89.9 (0.5)	80.6 (1.6)	89.8 (1.3)	86.7 (3.2)	9.7 (6.4)	0
Full-size	Fine-tuning	95.0	58.7	90.8	89.4	87.8	**97.0**	97.4	62.6	1
	Template Prompt	95.1	59.0	91.2	91.6	89.8	95.5	96.8	66.3	0
	w/ auto label word	95.6	58.8	91.7	91.3	90.8	95.8	97.0	**68.0**	1
	w/ special token	95.6	58.3	91.6	**92.7**	90.3	96.4	84.8	65.4	1
	Schema Prompt	95.2	**59.4**	91.4	91.8	90.2	95.6	97.2	67.0	1
	w/ auto label word	95.1	58.4	91.9	91.9	89.8	96.7	**97.6**	66.4	1
	w/ special token	**95.8**	57.7	**92.4**	91.4	**90.9**	96.4	97.2	67.8	3
	Null Prompt	95.7	55.9	90.5	87.5	90.8	96.0	96.4	**68.0**	1

Table 4: Experiment results on sentence classification. We report standard deviation for few-shot experiments. The results with *null* prompt (Logan IV et al., 2021) are produced by our re-implementation. The results with *null* prompts should be compared with template- and schema-based prompts without auto label word or special token.

	CoNLL03 (F1)		TACRED (F1)	
	Few	Full	Few	Full
Fine-tuning	34.3 (2.9)	90.8	20.8 (1.1)	64.9
Template	62.1 (5.1)	90.3	27.2 (1.6)	69.77
Schema	70.0 (2.9)	90.7	28.1 (1.6)	69.82

Table 5: Experiment results for named entity recognition and relation classification.

the *pattern* is entirely removed, and only the label word is utilized.

2.3.1 Label Word

For sentence classification, different types of label words are investigated, including manually-designed ones, automatically searched ones (Gao et al., 2021) and special tokens. The former two are pretrained words that are already in the vocabulary, while special tokens are those label tokens that are randomly initialized from scratch, with one token (e.g., `[T0]` and `[T1]`) for each label. We adopt the automatic label word searching method of Gao et al. (2021). Though other work has also investigated automatic label word generation (Schick et al., 2020), we do not include them all because the label word searching methodology is not our focus.

3 Analysis

3.1 Do Prompts Have to Follow Human Speaking And Writing?

The results for sentence classification are shown in Table 4. Under the few-shot setting, schema-based prompts achieve the best performance on all datasets except for CoLA. Taking different types of prompts into comparison, schemas consistently outperform templates by $0.6 \sim 2.2$ points and outperform *null* prompts by $0.5 \sim 4.6$ points (except for CoLA), indicating that schemas are better few-shot learners. Under the full-size setting, although the performance gaps, ranging from 0.1 to 0.7 points, are not as significant as those of the few-shot setting, schema-based prompts still gain 5 wins out of 8 datasets.

Table 5 gives the results of structure prediction tasks. Under the few-shot setting, schemas give the best performance, while prompting methods are generally better than fine-tuning. The improvements are much more significant than those of the sentence classification tasks, with absolute improvements of 35.7 F1 for CoNLL03 and 7.3 F1 for TACRED. Under the rich-resource setting, fine-tuning and prompting methods are shown to be competitive with each other.

Interestingly, we observe that the standard deviations of schema-based prompts are consistently the lowest among all methods, especially for comparing with template-based ones. Since standard deviation mainly results from the differences among randomly sampled train and development sets across 5 runs, the low standard deviation suggests that schema-based prompts are more stable than templates when data resources are limited. This also suggests that the performance lower-bound of schema-based prompts is relatively high, regardless of the quality of randomly sampled datasets. This advantage of schema-based prompts may be

Dataset	Labels	Label Words		
		Manual	**Automatic**	
		Template & Schema	Template	Schema
SST-2	positive / negative	great / terrible	exquisite / disgusting	$\boxed{\text{pure / dead}}$
MR	positive / negative	great / terrible	magical / laughable	$\boxed{\text{brilliant / blah}}$
Subj	subjective / objective	subjective / objective	obvious / murder	$\boxed{\text{Nil / unknown}}$

Table 6: Labels, manually-designed label words and automatically-searched label words for three datasets. We use the same set of manually-designed label words for both template and schema-based prompts. Those label words that obtain the best performance under the few-shot setting are $\boxed{\text{framed out}}$.

SST-2 (Label Words: great / terrible)	**Acc**
Manual:	
<Input> It was `[MASK].`	92.6 (0.6)
Auto Template:	
<Input> It's `[MASK]!`	92.7 (0.9)
<Input> That's `[MASK].`	92.6 (0.7)
<Input> Its `[MASK].`	92.4 (0.8)
It's `[MASK].` *<Input>*	92.1 (1.1)
Absolutely `[MASK].` *<Input>*	91.4 (1.4)
Just `[MASK].` *<Input>*	89.9 (1.6)

Table 7: Examples for automatically-searched templates and their performance on SST-2 dataset.

of great use in real-life scenarios, in which datasets often have very limited sizes and much noise.

As shown in Table 4, prompting methods with automatically-searched label words achieve the best results on 5 out of 8 datasets under the few-shot setting, showing their advantage against human-designed label words. In particular, all these 5 wins are obtained by schema-based prompts. Because the automatic label words are generated when the language model is not tuned yet, we conclude that pretrained LMs can already make use of grammatically-incorrect schemas even before any tuning. This implies that schemas might also be effective under zero-shot settings, which we leave for future investigation.

3.2 Are Task-Specific Hints Needed?

Few-shot Setting Though Logan IV et al. (2021) argued that *null* prompts could achieve competitive results compared with manually designed and automatically-generated prompts under the few-shot setting, our experiments give different results. Taking schema-based and *null* prompts into comparison[1], the former outperforms the latter for most few-shot tasks, with the only exception being CoLA on which the performance gap is just 0.1%. Schema-based and *null* prompts are mainly different in that the former is augmented with task-specific hints while the latter is not. Taking question classification task (TREC Dataset) as an example, the corresponding schema-based prompt is "*<Input>* Question type: `[MASK].`" while the *null* prompt is "*<Input>* `[MASK].`". An absolute improvement of 1.1 % is obtained by merely augmenting the prompt with two task-related words (i.e., "Question type"). Therefore, we can conclude that task-specific hints are still needed for prompt-based few-shot learning.

Rich-resource Setting Under the fully-supervised setting, schemas and *null* prompts are competitive with each other and either wins for 4 out of 8 datasets. The impacts of task-specific hints are overridden by large amounts of training data. Fine-tuning methods give an even more extreme condition: the hints and the label words are both removed, but fine-tuning still shows competitive performance with prompting. This is in line with previous work, which suggested that prompting methods are mostly effective for zero- and few-shot settings.

3.3 Automatic Search versus Manual

By examining all automatically searched label words, we find that most of them that obtain superior performance tend to be unnatural (e.g., "dead", "blah" and "Nil" as shown in Table 6[2]), which further verifies that prompts do not need to strictly follow the way in which humans speak and write. However, the automatic label words are not totally nonsense. For example, "dead" and "terrible" both tend to be negative, and "unknown" and "objective" both mean that something is out of one's mind. This "loosely-connected synonym" situation results in an assumption that effective label words should be consistent with human intuitions, though they are not strictly required to be natural or grammatically-correct.

We take SST-2 as an example to examine the effect of automatically-searched prompt templates. As shown in Table 7, the automatically generated template outperforms the manual one by adding a "!" at the end of the sentence, which suggests that automatic search is the optimal. Existing work that investigated automatic prompt search also pointed out that machine-generated prompts are superior to human-designed ones (Shin et al., 2020; Jiang et al., 2020; Gao et al., 2021).

[1]For fair comparison, we compare the "Schema Prompt" row (without automatic label words or special tokens) and the "*Null* Prompt" row in Table 4.

[2]The full list of label words are shown in Appendix A.2

References

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

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.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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.

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.

Robert L Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

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.

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*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

A Appendix

A.1 Detailed Experimental Settings

For sentence classification, our experiments are conducted on 8 datasets, including SST-2 (Socher et al., 2013), SST-5, MR, CR, MPQA, Subj (Pang and Lee, 2004), TREC (Hovy et al., 2001) and

Dataset	Labels	Label Words		
		Manual	Automatic	
		Template & Schema	Template	Schema
SST-2	positive / negative	great / terrible	exquisite / disgusting	pure / dead
SST-5	v.pos. / positive / neutral / negative / v.neg.	great / good / okay / bad / terrible	excellent / good / hilarious / terrible / awful	pure / appropriate / ok / low / dead
MR	positive / negative	great / terrible	magical / laughable	brilliant / blah
CR	positive / negative	great / terrible	astounding / worse	winning / boring
MPQA	positive / negative	great / terrible	awesome / awful	Good / FALSE
Subj	subjective / objective	subjective / objective	obvious / murder	Nil / unknown
TREC	description / entity / abbreviation / human / location / number	Description / Entity / Expression / Human / Location / Number	Discussion / Scene / Response / Fact / Results / Problem	Background / Static / Communication / Criminal / Location / Numbers
CoLA	grammatical / not_grammatical	correct / incorrect	fiction / now	c / N

Table 8: Labels, manually-designed label words and automatically-searched label words for each dataset. We use the same set of manually-designed label words for both template and schema-based prompts. Those label words that obtain the best performance under the few-shot setting are framed out.

CoLA (Warstadt et al., 2018). Similar to previous work, we adopt a masked language model to predict the label word and then adopt a verbalizer to map the label words to classification labels.

For structured prediction tasks, we adopt a pre-trained BART (Lewis et al., 2020) as Cui et al. (2021) did. As for the NER task, all possible text spans are enumerated as potential entity spans. The prompt sentences are generated by BART and the entity label is determined by comparing the summations of the log-likelihood of the generation process.

For the few-shot experiments, we choose $K = 16$, where K refers to the number of examples per class for the training and development sets. We conduct few-shot experiments across 5 runs, using different randomly sampled train and development sets. We randomly sample 2,000 examples as the test set for sentence classification and use the full test set for structure prediction.

A.2 Automatically Searched Label Words

The full list of automatically searched label words are shown in Table 8.