7.2.5 Prompt+LM Tuning

In this setting, there are prompt-relevant parameters, which can be fine-tuned together with the all or some of the parameters of the pre-trained models. Representative examples include PADA [8], P-Tuning [103]. Notably, this setting is very similar to the standard pre-train and fine-tune paradigm, but the addition of the prompt can provide additional bootstrapping at the start of model training.

- **Advantages**: This is the most expressive method, likely suitable for high-data settings.
- **Disadvantages**: Requires training and storing all parameters of the models. May overfit to small datasets.

8 Applications

In previous sections, we examined prompting methods from the point of view of the mechanism of the method itself. In this section, we rather organize prompting methods from the point of view of which applications they have been applied to. We list these applications in Tab. 7-8 and summarize them in the following sections.

8.1 Knowledge Probing

Factual Probing *Factual probing* (a.k.a. fact retrieval) is one of the earliest scenarios with respect to which prompting methods were applied. The motivation of exploring this task is to quantify how much factual knowledge the pre-trained LM's internal representations bear. In this task, parameters of pre-trained models are usually fixed, and knowledge is retrieved by transforming the original input into a cloze prompt as defined in §2.2, which can be manually crafted or automatically discovered. Relevant datasets including `LAMA` (Petroni et al., 2019) and `X-FACTR` (Jiang et al., 2020a). Since the answers are pre-defined, fact retrieval only focuses on finding effective templates and analyzing the results of different models using these templates. Both discrete template search (Petroni et al., 2019, 2020; Jiang et al., 2020c,a; Haviv et al., 2021; Shin et al., 2020; Perez et al., 2021) and continuous template learning (Qin and Eisner, 2021; Liu et al., 2021b; Zhong et al., 2021b) have been explored within this context, as well as prompt ensemble learning (Jiang et al., 2020c; Qin and Eisner, 2021).

Linguistic Probing Besides factual knowledge, large-scale pre-training also allows LMs to handle linguistic phenomena such as analogies (Brown et al., 2020), negations (Ettinger, 2020), semantic role sensitivity (Ettinger, 2020), semantic similarity (Sun et al., 2021), cant understanding (Sun et al., 2021), and rare word understanding (Schick and Schütze, 2020). The above knowledge can also be elicited by presenting *linguistic probing* tasks in the form of natural language sentences that are to be completed by the LM.

8.2 Classification-based Tasks

Prompt-based learning has been widely explored in classification-based tasks where prompt templates can be constructed relatively easily, such as text classification (Yin et al., 2019) and natural language inference (Schick and Schütze, 2021a). The key to prompting for classification-based tasks is reformulating it as an appropriate prompt. For example, Yin et al. (2019) use a prompt such as "the topic of this document is `[Z]`.", which is then fed into mask pre-trained LMs for slot filling.

Text Classification For *text classification* tasks, most previous work has used cloze prompts, and both prompt engineering (Gao et al., 2021; Hambardzumyan et al., 2021; Lester et al., 2021) and answer engineering (Schick and Schütze, 2021a; Schick et al., 2020; Gao et al., 2021) have been explored extensively. Most existing works explore the efficacy of prompt learning for text classification in the context of *few-shot* setting with "*fixed-prompt LM Tuning*" strategies (defined in §7.2.4).

Natural Language Inference (NLI) NLI aims to predict the relationship (e.g., `entailment`) of two given sentences. Similar to text classification tasks, for *natural language inference* tasks, cloze prompts are commonly used (Schick and Schütze, 2021a). Regarding prompt engineering, researchers mainly focus on the template search in the few-shot learning setting and the answer space $\mathcal{Z}$ is usually manually pre-selected from the vocabulary.

Work	Task	PLM	Setting	Prompt Engineering			Answer Engineering			Tuning	Mul-Pr
				Shape	Man	Auto	Shape	Man	Auto		
LMComm [173]	CR	L2R	Zero	Clo	✓	-	Sp	✓	-	TFP	-
GPT-2 [140]	CR,QA SUM,MT	GPT-2	Zero,Few	Clo,Pre	✓	-	Tok,Sp,Sen	✓	-	TFP	PA
WNLaMPro [150]	LCP	BERT	Zero	Clo	✓	-	Tok	✓	-	TFP	-
LMDiagnose [39]	CR,LCP	BERT	Zero	Clo	✓	-	Tok	✓	-	TFP	-
AdvTrigger [177]	GCG	GPT-2	Full	Pre	-	Disc	Sen	✓	-	TFP	-
CohRank [31]	CKM	BERT	Zero	Clo	✓	-	Tok,Sp	✓	-	TFP	-
LAMA [133]	FP	Conv,Trans ELMo,BERT	Zero	Clo	✓	-	Tok	✓	-	TFP	-
CTRL [75]	GCG	CTRL	Full	Pre	✓	-	Sen	✓	-	LMT	-
T5 [141]	TC,SUM QA,MT	T5	Full	Pre	✓	-	Tok,Sp,Sen	✓	-	LMT	-
Neg & Mis [74]	FP	Trans,ELMo BERT	Zero	Clo	✓	-	Tok	✓	-	TFP	-
LPAQA [68]	FP	BERT,ERNIE	Full	Clo	✓	Disc	Tok	✓	-	TFP	PE
ZSC [135]	TC	GPT-2	Full	Pre	✓	-	Tok,Sp	✓	-	LMT	-
PET-TC [153]	TC	RoBERTa,XLM-R	Few	Pre	✓	-	Tok	✓	Disc	LMT	PE
ContxFP [132]	FP	BERT,RoBERTa	Zero	Clo	✓	Disc	Tok	✓	-	TFP	-
UnifiedQA [76]	QA	T5,BART	Full	Prefix	✓	-	Tok,Sp,Sen	✓	-	LMT	-
RAG [95]	QA,GCG,TC	BART	Full	Pre	-	Disc	Tok,Sp,Sen	✓	-	LMPT	PE
GPT-3 [16]	QA,MT,GCG CR,TC,LCP MR,SR,AR	GPT-3	Zero,Few	Clo,Pre	✓	-	Tok,Sp,Sen	✓	-	TFP	PA
CommS2S [187]	CR	T5	Full	Pre	✓	-	Tok	✓	-	LMT	-
PET-SGLUE [154]	TC	ALBERT	Few	Clo	✓	-	Tok,Sp	✓	-	LMT	PE
ToxicityPrompts [47]	GCG	GPT-1,GPT-2 GPT-3,CTRL	Zero	Pre	✓	-	N/A			TFP	-
WhyLM [147]	Theory	GPT-2	Full	Pre	✓	-	Tok	✓	-	PT	-
X-FACTR [66]	FP	mBERT,BERT XLM,XLM-R	Zero	Clo	✓	-	Tok,Sp	✓	-	TFP	-
Petal [149]	TC	RoBERTa	Few	Clo	✓	-	Tok	-	Disc	LMT	PE
AutoPrompt [159]	TC,FP,IE	BERT,RoBERTa	Full	Clo	-	Disc	Tok	-	Disc	TFP	-
CTRLsum [59]	SUM	BART	Full	Pre	✓	-	Sen	✓	-	LMT	-
PET-Gen [152]	SUM	PEGASUS	Few	Pre	✓	-	Sen	✓	-	LMT	PE
LM-BFF [46]	TC	RoBERTa	Few	Clo	-	Disc	Tok	-	Disc	LMT	PE,PA
WARP [55]	TC	RoBERTa	Few,Full	Clo,Pre	✓	Cont	Tok	✓	Cont	PT	PE
Prefix-Tuning [96]	D2T,SUM	GPT-2,BART	Full	Pre	-	Cont	Sen	✓	-	PT	-
KATE [100]	TC,D2T,QA	GPT-3	Few	Pre	✓	-	Tok,Sp,Sen	✓	-	TFP	PA
PromptProg [145]	MT,MR AR,QA	GPT-3	Zero,Few	Pre	✓	-	Tok,Sp,Sen	✓	-	TFP	PA
ContxCalibrate [201]	TC,FP,IE	GPT-2,GPT-3	Few	Pre	✓	-	Tok,Sp	✓	-	TFP	PA
PADA [8]	TC,TAG	T5	Full	Pre	-	Disc	N/A			LMPT	-
SD [155]	GCG	GPT-2	Zero	Pre	✓	-	N/A			TFP	-
BERTese [58]	FP	BERT	Full	Clo	✓	Disc	Tok	✓	-	TFP	-
Prompt2Data [148]	TC	RoBERTa	Full	Clo	✓	-	Tok,Sp	✓	-	LMT	-
P-Tuning [103]	FP,TC	GPT-2,BERT ALBERT	Few,Full	Clo,Pre	✓	Cont	Tok,Sp	✓	-	TFP,LMPT	-
GLM [37]	TC	GLM	Full	Clo	✓	-	Tok,Sp	✓	-	LMT	-

Table 7: An organization of works on prompting (Part 1). See the caption of Tab. 8 for a detailed description for all the abbreviations used in this table.

Work	Task	PLM	Setting	Prompt Engineering			Answer Engineering			Tuning	Mul-Pr
				Shape	Man	Auto	Shape	Man	Auto		
ADAPET [170]	TC	ALBERT	Few	Clo	✓	-	Tok,Sp	✓	-	LMT	-
Meta [202]	TC	T5	Full	Pre	✓	-	Tok	✓	-	LMT	-
OptiPrompt [203]	FP	BERT	Full	Clo	✓	Cont	Tok	✓	-	PT	-
Soft [137]	FP	BERT,BART RoBERTa	Full	Clo	✓	Cont	Tok	✓	-	PT	PE
DINO [151]	GCG	GPT-2	Zero	Pre	✓	-	N/A			TFP	-
AdaPrompt [21]	IE	BERT	Few,Full	Clo	✓	-	Tok	-	Disc	LMT	-
PMI_{DC} [62]	GCG,QA,TC	GPT-2,GPT-3	Zero	Pre	✓	-	Tok,Sp,Sen	✓	-	TFP	-
Prompt-Tuning [91]	TC	T5	Full	Pre	-	Cont	Tok,Sp	✓	-	PT	PE
Natural-Instr [120]	GCG	GPT-3,BART	Few,Full	Pre	✓	-	Tok,Sp,Sen	✓	-	TFP,LMT	PA
OrderEntropy [111]	TC	GPT-2,GPT-3	Few	Pre	✓	-	Tok	✓	-	TFP	PA
FewshotSemp [158]	SEMP	GPT-3	Few	Pre	✓	-	Sen	✓	-	TFP	PA
PanGu-α [194]	QA,CR,TC SUM,GCG	PanGu-α	Zero,Few	Clo,Pre	✓	-	Tok,Sp,Sen	✓	-	TFP	PA
TrueFewshot [129]	TC,FP	GPT-2,GPT-3 ALBERT	Few	Clo,Pre	✓	Disc	Tok,Sp	✓	-	TFP,LMT	-
PTR [56]	IE	RoBERTa	Full	Clo	✓	Cont	Tok,Sp	✓	-	LMPT	PC
TemplateNER [29]	TAG	BART	Few,Full	Clo,Pre	✓	-	Tok	✓	-	LMT	PD
PERO [83]	TC,FP	BERT,RoBERTa	Few	Pre	✓	-	Tok	✓	-	TFP	PA
PromptAnalysis [181]	Theory	BERT	Full	Clo	-	Cont	N/A			PT	-
CPM-2 [198]	QA,MR,SUM TC,GCG,MT	CPM-2	Full	Pre	-	Cont	Tok,Sp,Sent	✓	-	PT,LMPT	-
BARTScore [193]	EVALG	BART	Zero	Pre	✓	Disc	Sen	✓	-	TFP	PE
NullPrompt [109]	TC	RoBERTa,ALBERT	Few	Pre	✓	-	Tok	✓	-	LMPT	-
Frozen [174]	VQA,VFP,MG	GPT-like	Full	Pre	-	Cont	Sp (Visual)	✓	-	PT	PA
ERNIE-B3 [167]	TC,LCP,NLI CR,QA,SUM GCG	ERNIE-B3	Zero	Clo,Pre	✓	-	Tok,Sp,Sen	✓	-	TFP	-
Codex [20]	CodeGen	GPT	Zero,Few Full	Pre	✓	-	Span	✓	Disc	TFP,LMT	PA
HTLM [1]	TC,SUM	BART	Zero,Few Full	Clo	✓	Disc	Tok,Sp,Sen	✓	-	LMT	PA
FLEX [15]	TC	T5	Zero,Few	Pre	✓	-	Tok,Sp	✓	-	LMT	-

Table 8: An organization of works on prompting (Part 2). The **Task** column lists the tasks that are performed in corresponding papers. We use the following abbreviations. **CR**: Commonsense Reasoning. **QA**: Question Answering. **SUM**: Summarization. **MT**: Machine Translation. **LCP**: Linguistic Capacity Probing. **GCG**: General Conditional Generation. **CKM**: Commonsense Knowledge Mining. **FP**: Fact Probing. **TC**: Text Classification. **MR**: Mathematical Reasoning. **SR**: Symbolic Reasoning. **AR**: Analogical Reasoning. **Theory**: Theoretical Analysis. **IE**: Information Extraction. **D2T**: Data-to-text. **TAG**: Sequence Tagging. **SEMP**: Semantic Parsing. **EVALG**: Evaluation of Text Generation. **VQA**: Visual Question Answering. **VFP**: Visual Fact Probing. **MG**: Multimodal Grounding. **CodeGen**: Code generation. The **PLM** column lists all the pre-trained LMs that have been used in corresponding papers for downstream tasks. **GPT-like** is an autoregressive language model which makes small modifications to the original GPT-2 architecture. For other pre-trained LMs, please refer to §3 for more information. **Setting** column lists the settings for prompt-based learning, can be zero-shot learning (**Zero**), few-shot learning (**Few**), fully supervised learning (**Full**). Under **Prompt Engineering**, **Shape** denotes the shape of the template (**Clo** for cloze and **Pre** for prefix), **Man** denotes whether human effort is needed, **Auto** denotes data-driven search methods (**Disc** for discrete search, **Cont** for continuous search). Under **Answer Engineering**, **Shape** indicates the shape of the answer (**Tok** for token-level, **Sp** for span-level, **Sen** for sentence- or document-level), and **Man** and **Auto** are the same as above. The **Tuning** column lists tuning strategies (§7). **TFP**: Tuning-free Prompting. **LMT**: Fixed-prompt LM Tuning. **PT**: Fixed-LM Prompt Tuning. **LMPT**: LM+Prompt Tuning. The **Mul-Pr** column lists multi-prompt learning methods. **PA**: Prompt Augmentation. **PE**: Prompt Ensembling. **PC**: Prompt Composition. **PD**: Prompt Decomposition.

8.3 Information Extraction

Unlike classification tasks where cloze questions can often be intuitively constructed, for *information extraction* tasks constructing prompts often requires more finesse.

Relation Extraction *Relation extraction* is a task of predicting the relation between two entities in a sentence. Chen et al. (2021b) first explored the application of *fixed-prompt LM Tuning* in relation extraction and discuss two major challenges that hinder the direct inheritance of prompting methodology from classification tasks: (1) The larger label space (e.g. 80 in relation extraction v.s 2 in binary sentiment classification) results in more difficulty in answer engineering. (2) In relation extraction, different tokens in the input sentence may be more or less important (e.g. entity mentions are more likely to participate in a relation), which, however, can not be easily reflected in the prompt templates for classification since the original prompt template regards each word equally. To address the above problems, Chen et al. (2021b) propose an adaptive answer selection method to address the issue (1) and task-oriented prompt template construction for the issue (2), where they use special markers (e.g. `[E]`) to highlight the entity mentions in the template. Similarly, Han et al. (2021) incorporate entity type information via multiple prompt composition techniques (illustrated in Fig. 4).

Semantic Parsing *Semantic parsing* is a task of generating a structured meaning representation given a natural language input. Shin et al. (2021) explore the task of few-shot semantic parsing using LMs by (1) framing the semantic parsing task as a paraphrasing task (Berant and Liang, 2014) and (2) constraining the decoding process by only allowing output valid according to a grammar. They experiment with the *in-context learning* setting described in §7.2.2, choosing answered prompts that are semantically close to a given test example (determined by the conditional generation probability of generating a test sample given another training example). The results demonstrate the effectiveness of the paraphrasing reformulation for semantic parsing tasks using pre-trained LMs.

Named Entity Recognition *Named entity recognition* (NER) is a task of identifying named entities (e.g., person name, location) in a given sentence. The difficulty of prompt-based learning's application to tagging tasks, exemplified as NER, is that, unlike classification, (1) each unit to be predicted is a token or span instead of the whole input text, (2) there is a latent relationship between the token labels in the sample context. Overall, the application of prompt-based learning in tagging task has not been fully explored. Cui et al. (2021) recently propose a template-based NER model using BART, which enumerates text spans and considers the generation probability of each type within manually crafted templates. For example, given an input "Mike went to New York yesterday", to determine what type of entity "Mike" is, they use the template "Mike is a `[Z]` entity", and the answer space $\mathcal{Z}$ consists of values such as "person" or "organization".

8.4 "Reasoning" in NLP

There is still a debate[6] about if deep neural networks are capable of performing "reasoning" or just memorizing patterns based on large training data (Arpit et al., 2017; Niven and Kao, 2019). As such, there have been a number of attempts to probe models' reasoning ability by defining benchmark tasks that span different scenarios. We detail below how prompting methods have been used in these tasks.

Commonsense Reasoning There are a number of benchmark datasets testing commonsense reasoning in NLP systems (Huang et al., 2019; Rajani et al., 2019; Lin et al., 2020; Ponti et al., 2020). Some commonly attempted tasks involve solving Winograd Schemas (Levesque et al., 2012), which require the model to identify the antecedent of an ambiguous pronoun within context, or involve completing a sentence given multiple choices. For the former, an example could be "The trophy doesn't fit into the brown suitcase because it is too large." And the task for the model is to infer whether "it" refers to the trophy or the "suitcase". By replacing "it" with its potential candidates in the original sentences and calculating the probability of the different choices, pre-trained LMs can perform quite well by choosing the choice that achieves the highest probability (Trinh and Le, 2018). For the latter, an example could be "Eleanor offered to fix her visitor some coffee. Then she realized she didn't have a clean `[Z]`.". The candidate choices are "cup", "bowl" and "spoon". The task for the pre-trained LM is to choose the one from the three candidates that most conforms to common sense. For these kinds of tasks, we can also score the generation probability of each candidate and choose the one with the highest probability (Ettinger, 2020).

Mathematical Reasoning Mathematical reasoning is the ability to solve mathematical problems, e.g. arithmetic addition, function evaluation. Within the context of pre-trained LMs, researchers have found that pre-trained embeddings and LMs can perform simple operations such as addition and subtraction when the number of digits is small, but fail when the numbers are larger (Naik et al., 2019; Wallace et al., 2019b; Brown et al., 2020). Reynolds and McDonell (2021) explore more complex mathematical (e.g. $f(\boldsymbol{x}) = \boldsymbol{x} * \boldsymbol{x}$, what is $f(f(3))$?) reasoning problems and improve LM performance through serializing reasoning for the question.

[6] e.g. https://medium.com/reconstruct-inc/the-golden-age-of-computer-vision-338da3e471d1

8.5 Question Answering

Question answering (QA) aims to answer a given input question, often based on a context document. QA can take a variety of formats, such as extractive QA (which identifies content from the context document containing the answer; e.g. SQuAD (Rajpurkar et al., 2016)), multiple-choice QA (where the model has to pick from several choices; e.g. RACE (Lai et al., 2017)), and free-form QA (where the model can return an arbitrary textual string as a response; e.g. NarrativeQA (Kočiský et al., 2018)). Generally, these different formats have been handled using different modeling frameworks. One benefit of solving QA problems with LMs, potentially using prompting methods, is that different formats of QA tasks can be solved within the same framework. For example, Khashabi et al. (2020) reformulate many QA tasks as a text generation problem by fine-tuning seq2seq-based pre-trained models (e.g. T5) and appropriate prompts from the context and questions. Jiang et al. (2020b) take a closer look at such prompt-based QA systems using sequence to sequence pre-trained models (T5, BART, GPT2) and observe that probabilities from these pre-trained models on QA tasks are not very predictive of whether the model is correct or not.

8.6 Text Generation

Text generation is a family of tasks that involve generating text, usually conditioned on some other piece of information. Prompting methods can be easily applied to these tasks by using *prefix prompts* together with autoregressive pre-trained LMs. Radford et al. (2019) demonstrated impressive ability of such models to perform generation tasks such as text summarization and machine translation using prompts such as "translate to french, `[X]`, `[Z]`". Brown et al. (2020) perform *in-context learning* (§7.2.2) for text generation, creating a prompt with manual templates and augmenting the input with multiple *answered prompts*. Schick and Schütze (2020) explore *fixed-prompt LM tuning* (§7.2.4) for few-shot text summarization with manually crafted templates. (Li and Liang, 2021) investigate *fixed-LM prompt tuning* (§7.2.3) for text summarization and data-to-text generation in few-shot settings, where learnable prefix tokens are prepended to the input while parameters in pre-trained models are kept frozen. Dou et al. (2021) explored the *prompt+LM tuning* strategy (§7.2.5) on text summarization task, where learnable prefix prompts are used and initialized by different types of guidance signals, which can then be updated together with parameters of pre-trained LMs.

8.7 Automatic Evaluation of Text Generation

Yuan et al. (2021b) have demonstrated that prompt learning can be used for automated evaluation of generated texts. Specifically, they conceptualize the evaluation of generated text as a text generation problem, modeled using a pre-trained sequence-to-sequence, and then use *prefix prompts* that bring the evaluation task closer to the pre-training task. They experimentally find that simply adding the phrase "such as" to the translated text when using pre-trained models can lead to a significant improvement in correlation on German-English machine translation (MT) evaluation.

8.8 Multi-modal Learning

Tsimpoukelli et al. (2021) shift the application of prompt learning from text-based NLP to the *multi-modal* setting (vision and language). Generally, they adopt the *fixed-LM prompt tuning* strategy together with *prompt augmentation* techniques. They specifically represent each image as a sequence of continuous embeddings, and a pre-trained LM whose parameters are frozen is prompted with this prefix to generate texts such as image captions. Empirical results show few-shot learning ability: with the help of a few demonstrations (answered prompts), system can rapidly learn words for new objects and novel visual categories.

8.9 Meta-Applications

There are also a number of applications of prompting techniques that are not NLP tasks in and of themselves, but are useful elements of training strong models for any application.

Domain Adaptation Domain adaptation is the practice of adapting a model from one domain (e.g. news text) to another (e.g. social media text). Ben-David et al. (2021) use self-generated *domain related features* (DRFs) to augment the original text input and perform sequence tagging as a sequence-to-sequence problem using a seq2seq pre-trained model.

Debiasing Schick et al. (2021) found that LMs can perform self-diagnosis and self-debiasing based on biased or debiased instructions. For example, to self-diagnosis whether the generated text contains violent information, we can use the following template "The following text contains violence. `[X][Z]`". Then we fill `[X]` with the input text and look at the generation probability at `[Z]`, if the probability of "Yes" is greater than "No", then we would assume the given text contains violence, and vice versa. To perform debiasing when generating text, we first compute the probability of the next word $P(x_t|\boldsymbol{x}_{<t};\theta)$ given the original input. Then we compute the probability

of next word $P(x_t|[\boldsymbol{x}_{<t}; \boldsymbol{x}_{\text{diagnosis}}]; \theta)$ by appending self-diagnosis textual input to the original input as mentioned above. These two probability distributions for the next token can be combined to suppress the undesired attribute.

Dataset Construction Schick and Schütze (2021) propose to use pre-trained LMs to generate datasets given certain instructions. As an example, suppose we have an unlabeled dataset in which each sample is a sentence. If we want to construct a dataset containing pairs of semantically similar sentences, then we can use the following template for each input sentence: "Write two sentences that mean the same thing. `[X] [Z]`" and attempt to generate a sentence that shares the same meaning as the input sentence.

8.10 Resources

We also collect some useful resources for different prompt-based applications.

Dataset Some datasets specifically designed for few-shot and zero-shot learning are shown in Tab. 9.

Task	Dataset	Setting	URL
Commonsense Reasoning	Pronoun Disambiguation Problems [93]	Zero	https://cs.nyu.edu/ davise/papers/...
	Winograd Schema Challenge [93]	Zero	https://cs.nyu.edu/ davise/papers/...
	CPRAG-102 [39]	Zero	https://github.com/aetting/lm-diagnostics
Linguistic Capacity Probing	WNLaMPro [150]	Zero	https://github.com/timoschick/...
	ROLE-88 [39]	Zero	https://github.com/aetting/lm-diagnostics
	NEG-136 [39]	Zero	https://github.com/aetting/lm-diagnostics
Fact Probing	LAMA [133]	Zero	https://dl.fbaipublicfiles.com/LAMA/...
	Negated LAMA [74]	Zero	https://github.com/norakassner/LAMA...
	Misprimed LAMA [74]	Zero	https://github.com/norakassner/LAMA...
	X-FACTR [66]	Zero	https://x-factr.github.io/
	LAMA-TREx-easy-hard [203]	Zero	https://github.com/princeton-nlp/...
Text Classification	FLEX [15]	Zero,Few	https://github.com/allenai/flex
	FewGLUE [154]	Few	https://github.com/timoschick/fewglue
General Conditional Gen.	REALTOXICITYPROMPTS [47]	Zero	https://allenai.org/data/...
	Natural-Instructions [120]	Few,Full	https://instructions.apps.allenai.org/

Table 9: Few-shot and zero-shot datasets for prompt-based learning.

Prompts As shown in Tab. 10, we collect existing commonly-used prompts designed manually, which can be regarded as off-the-shelf resource for future research and applications.

9 Prompt-relevant Topics

What is the essence of prompt-based learning and how does it relate to other learning methods? In this section, we connect prompt learning with other similar learning methods.

Ensemble Learning *Ensemble learning* (Ting and Witten, 1997; Zhou et al., 2002) is a technique that aims to improve the performance of a task by taking advantage of the complementarity of multiple systems. Generally, the different systems used in an ensemble result from different choices of architectures, training strategies, data ordering, and/or random initialization. In prompt ensembling (§6.1), the choice of prompt templates becomes another way to generate multiple results to be combined. This has the clear advantage that this does not necessarily require training the model multiple times. For example, when using discrete prompts, these prompts can simply be changed during the inference stage (Jiang et al., 2020c).

Few-shot Learning *Few-shot learning* aims to learn a machine learning system in the data-scarce scenarios with few training samples. There are a wide variety of methods to achieve few-shot learning including model agnostic meta-learning (Finn et al., 2017b) (learning features rapidly adaptable to new tasks), embedding learning (Bertinetto et al., 2016) (embedding each sample in a lower-dimensional space where similar samples are close together), memory-based learning (Kaiser et al., 2017) (representing each sample by a weighted average of contents from the memory) etc. (Wang et al., 2020). Prompt augmentation can be regarded as another way to achieve few-shot learning (a.k.a. priming-based few-shot learning (Kumar and Talukdar, 2021)). Compared to previous methods, prompt augmentation directly prepends several labeled samples to the currently-processed sample elicit knowledge from pre-trained LMs even without any parameter tuning.