et al., 2020) and text classification (Chai et al., 2020). These methods are very similar to the prompting methods introduced here in that they use textual questions to specify which task is to be performed. However, one of the key points of prompting methods is how to better use the knowledge in pre-trained LMs, and these were not covered extensively on previous works advocating for QA formulations.

Controlled Generation *Controlled generation* aims to incorporate various types of guidance beyond the input text into the generation model (Yu et al., 2020). Specifically, the guidance signals could be *style tokens* (Sennrich et al., 2016b; Fan et al., 2018), *length specifications* (Kikuchi et al., 2016), *domain tags* (Chu et al., 2017), or any variety of other pieces of information used to control of the generated text. It could also be *keywords* (Saito et al., 2020), *relation triples* (Zhu et al., 2020) or even *highlighted phrases or sentences* (Grangier and Auli, 2018; Liu et al., 2021c) to plan the content of generated texts. In a way, many of the prompting methods described here are a type of controllable generation, where the prompt is usually used to specify the *task itself*. Thus, it is relatively easy to find commonalities between the two genres: (1) both add extra information to the input text for better generation, and these additional signals are (often) learnable parameters. (2) If "controlled generation" is equipped with seq2seq-based pre-trained models (e.g., BART), then it is can be regarded as prompt learning with input-dependent prompts and the *prompt+LM fine-tuning* strategy (§7.2.5), e.g. *GSum* (Dou et al., 2021), where both the prompt's and pre-trained LM's parameters can be tuned.

Also, some clear discrepancies between controlled generation and prompt-based text generation are: (1) In controlled generation work, the control is generally performed over the style or content of the generations (Fan et al., 2018; Dou et al., 2021) while the underlying task remains the same. They don't necessarily require a pre-trained model. In contrast, the main motivation for using prompts for text generation is to specify the task itself and better utilize the pre-trained model. (2) Moreover, most of the current work on prompt learning in text generation shares a dataset- or task-level prompt (Li and Liang, 2021). Only very few works have explored input-dependent ones (Tsimpoukelli et al., 2021). However, this is a common setting and effective in the controlled text generation, which may provide valuable direction for the future work on prompt learning.

Supervised Attention Knowing to pay attention to the important information is a key step when extracting useful information from objects such as long text sequences (Liu et al., 2016; Sood et al., 2020), images (Sugano and Bulling, 2016; Zhang et al., 2020b), or knowledge bases (Yu et al., 2020; Dou et al., 2021)). *Supervised attention* (Liu et al., 2017b) aims to provide explicit supervision over the attention of models based on the fact that completely data-driven attention can overfit to some artifacts (Liu et al., 2017a). In this respect, prompt learning and supervised attention share ideas that both aim to extract salient information with some clues, which need to be provided separately. To solve this problem, supervised attention methods tried to use additional loss functions to learn to predict gold attention on a manually labeled corpus (Jiang et al., 2015; Qiao et al., 2018; Gan et al., 2017). Research on prompt learning may also borrow ideas from this literature.

Data Augmentation Data augmentation is a technique that targets increasing the amount of data that can be used for training by making modifications to existing data (Fadaee et al., 2017; Ratner et al., 2017). As recently observed by (Scao and Rush, 2021), adding prompts can achieve a similar accuracy improvement to the addition of 100s of data points on average across classification tasks, which suggests that using prompts for a downstream task is similar to conducting data augmentation implicitly.

10 Challenges

Although prompt-based learning has shown significant potential among different tasks and scenarios, several challenges remain, some of which we detail below.

10.1 Prompt Design

Tasks beyond Classification and Generation Most existing works about prompt-based learning revolve around either text classification or generation-based tasks. Applications to information extraction and text analysis tasks have been discussed less, largely because the design of prompts is less straightforward. We expect that applying prompting methods to these tasks in the future it will require either reformulating these tasks so that they can be solved using classification or text generation-based methods, or performing effective answer engineering that expresses structured outputs in an appropriate textual format.

Prompting with Structured Information In many NLP tasks, the inputs are imbued with some variety of structure, such as tree, graph, table, or relational structures. How to best express these structures in prompt or answer engineering is a major challenge. Existing works (Chen et al., 2021b) make a step by making prompts with additional marks to encode lexical information, such as entity markings. Aghajanyan et al. (2021) present structured prompts based on hyper text markup language for more fine-grained web text generation. However, moving beyond this to more complicated varieties of structure is largely unexplored, and a potentially interesting research area.

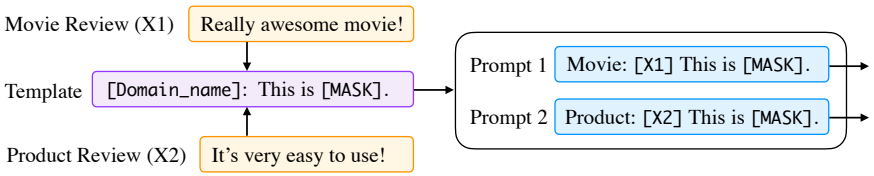

Figure 5: Multi-prompt learning for multi-task, multi-domain or multi-lingual learning. We use different colors to differentiate different components as follows. " " for input text, " " for template, " " for prompt.

Entanglement of Template and Answer The performance of a model will depend on *both* the templates being used and the answer being considered. How to simultaneously search or learn for the best combination of template and answer remains a challenging question. Current works typically select answers before select template (Gao et al., 2021; Shin et al., 2020), but Hambardzumyan et al. (2021) have demonstrated the initial potential of simultaneously learning both.

10.2 Answer Engineering

Many-class and Long-answer Classification Tasks For classification-based tasks, there are two main challenges for answer engineering: (a) When there are too many classes, how to select an appropriate answer space becomes a difficult combinatorial optimization problem. (b) When using multi-token answers, how to best decode multiple tokens using LMs remains unknown, although some multi-token decoding methods have been proposed (Jiang et al., 2020a).

Multiple Answers for Generation Tasks For text generation tasks, qualified answers can be semantically equivalent but syntactically diverse. So far, almost all works use prompt learning for text generation relying solely on a single answer, with only a few exceptions (Jiang et al., 2020c). How to better guide the learning process with multiple references remains a largely open research problem.

10.3 Selection of Tuning Strategy

As discussed in §7, there are a fairly wide variety of methods for tuning parameters of prompts, LMs, or both. However, given the nascent stage of this research field, we still lack a systematic understanding of the tradeoffs between these methods. The field could benefit from systematic explorations such as those performed in the pre-train and fine-tune paradigm regarding the tradeoffs between these different strategies (Peters et al., 2019).

10.4 Multiple Prompt Learning

Prompt Ensembling In prompt ensembling methods, the space and time complexity increase as we consider more prompts. How to distill the knowledge from different prompts remains underexplored. Schick and Schütze (2020, 2021a,b) use an ensemble model to annotate a large dataset to distill the knowledge from multiple prompts.

In addition, how to select ensemble-worthy prompts is also under-explored. For text generation tasks, the study of prompt ensemble learning has not been performed so far, probably because ensemble learning in text generation itself is relatively complicated. To remedy this problem, some recently proposed neural ensembling methods such as *Refactor* (Liu et al., 2021c) could be considered as a method for prompt ensembling in text generation tasks.

Prompt Composition and Decomposition Both prompt composition and decomposition aim to break down the difficulty of a complicated task input by introducing multiple sub-prompts. In practice, how to make a good choice between them is a crucial step. Empirically, for those token (Ma and Hovy, 2016) or span (Fu et al., 2021) prediction tasks (e.g., NER), prompt decomposition can be considered, while for those span relation prediction (Lee et al., 2017) tasks (e.g., entity coreference), prompts composition would be a better choice. In the future, the general idea of de-/composing can be explored in more scenarios.

Prompt Augmentation Existing prompt augmentation methods are limited by the input length, i.e., feeding too many demonstrations to input is infeasible. Therefore, how to select informative demonstrations, and order them in an appropriate is an interesting but challenging problem (Kumar and Talukdar, 2021).

Prompt Sharing All the above considerations refer to the application of prompt in a single task, domain or language. We may also consider *prompt sharing*, where prompt learning is applied to multiple tasks, domains, or languages. Some key issues that may arise include how to design individual prompts for different tasks, and how to modulate their interaction with each other. So far this field has not been explored. Fig.5 illustrates a simple multiple prompt learning strategy for multiple tasks, where prompt templates are partially shared.

10.5 Selection of Pre-trained Models

With plenty of pre-trained LMs to select from (see §3), how to choose them to better leverage prompt-based learning is an interesting and difficult problem. Although we have conceptually introduced (§3.4) how different paradigms of pre-trained models are selected for diverse NLP tasks, there are few to no systematic comparisons of the benefits brought by prompt-based learning for different pre-trained LMs.

10.6 Theoretical and Empirical Analysis of Prompting

Despite their success in many scenarios, theoretical analysis and guarantees for prompt-based learning are scarce. Wei et al. (2021) showed that soft-prompt tuning can relax the non-degeneracy assumptions (the generation probability of each token is linearly independent) needed for downstream recovery (i.e. recover the ground-truth labels of the downstream task.), making it easier to extract task-specific information. Saunshi et al. (2021) verified that text classification tasks can be reformulated as sentence completion tasks, thus making language modeling a meaningful pre-training task. Scao and Rush (2021) empirically show that prompting is often worth 100s of data points on average across classification tasks.

10.7 Transferability of Prompts

Understanding the extent to which prompts are specific to the model and improving the transferability of prompts are also important topics. (Perez et al., 2021) show that prompts selected under tuned few-shot learning scenario (where one has a larger validation set to choose prompts) generalize well across models of similar sizes while prompts selected under true few-shot learning scenario (where one only has a few training samples) do not generalize as effectively as the former setting among models with similar sizes. The transferability is poor when the model sizes are quite different in both scenarios.

10.8 Combination of Different Paradigms

Notably, much of the success of the prompting paradigm is built on top of pre-trained models that were developed for the pre-train and fine-tune paradigm, such as BERT. However, are the pre-training methods that are effective for the latter applicable as-is to the former, or can we entirely re-think our pre-training methods to further improve accuracy or ease of applicability to prompting-based learning? This is an important research question that has not been covered extensively by the literature.

10.9 Calibration of Prompting Methods

Calibration (Gleser, 1996) refers to the ability of a model to make good probabilistic predictions. When using the generation probability of the pre-trained LMs (e.g., BART) to predict the answer, we need to be careful since the probability distribution is typically not well calibrated. Jiang et al. (2020b) observed the probabilities of pre-trained models (e.g., BART, T5, GPT-2) on QA tasks are well calibrated. Zhao et al. (2021) identify three pitfalls (majority label bias, recency bias and common token bias) that lead the pre-trained LMs to be biased toward certain answers when provided answered prompts. For example, if the final answered prompt has a positive label, then this will bias the model towards predicting positive words. To overcome those pitfalls, Zhao et al. (2021) first use context-free input (e.g. the prompt would be "Input: Subpar acting. Sentiment: Negative\n Input: Beautiful film. Sentiment: Positive\n Input: N/A. Sentiment:") to get the initial probability distribution P_0, then they use the real input (e.g. the prompt would be "Input: Subpar acting. Sentiment: Negative\n Input: Beautiful film. Sentiment: Positive\n Input: Amazing. Sentiment:") to get the probability distribution P_1. Finally, these two distributions can be used to get a calibrated generation probability distribution. However, this method has two drawbacks: (1) it comes with the overhead of finding proper context-free input (e.g. whether to use "N/A" or "None") and (2) the probability distribution of the underlying pre-trained LM is still not calibrated.

Even though we have a calibrated probability distribution, we also need to be careful when we assume a single gold answer for an input. This is because that all surface forms of a same object will compete for finite probability mass (Holtzman et al., 2021). For example, if we consider the gold answer to be "Whirlpool bath", the generation probability of it will typically be low since the word "Bathtub" shares the same meaning and it will take over a large probability mass. To address this issue, we could either (i) perform answer engineering to construct a comprehensive gold answer set using paraphrasing methods (§5.2.2) or (ii) calibrate the probability of a word based on its prior likelihood within the context (Holtzman et al., 2021).

11 Meta Analysis

In this section, we aim to give a quantitative birds-eye view of existing research on prompting methods by performing a meta analysis over existing research works along different dimensions.