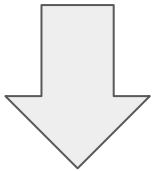
Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

Link : https://arxiv.org/pdf/2107.13586.pdf

Presented by: Ayush Koirala

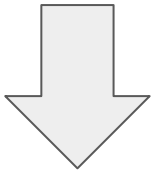
History In NLP Technical Development

Feature Engineering



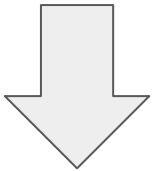
- Supervised Learning (Non-neural network)
- Mostly popular in 2015
- **Characteristics:**
  - Non-neural machine learning is used
  - Required manually defined feature extraction
- **Representation work:**
  - Manual features: Linear or kernelized **SVM**
  - **Conditional random fields(CRF)**

Architecture Engineering



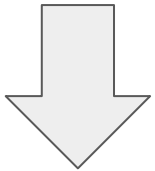
- Supervised Learning (neural network)
- Popular 2013-2018
- **Characteristics**:
  - Rely on neural network
  - No manually define features (LSTM, CNN)
- **Representation work:**
  - **CNN for Text classification**

Objective Engineering



- Pre-train, Fine-tune
- popular in 2017 - Now
- **Characteristics:**
  - Pretrain LMs used as initialization of full model - both shallow and deep features
  - More work on engineering objective functions
- **Representation work:**
  - **Bert → Fine Tuning**

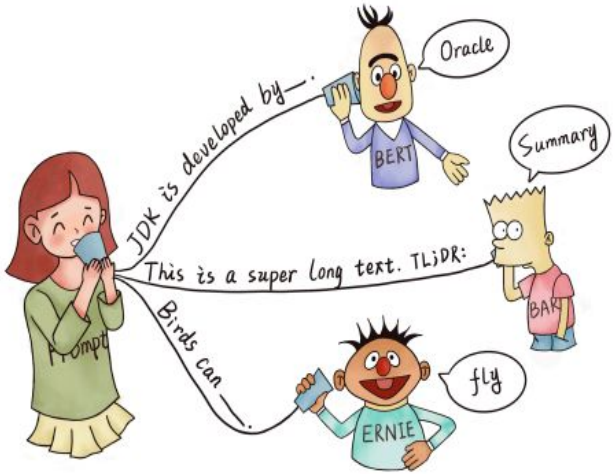
Prompt Engineering



- Pre-train, prompt, Predict
- 2019-Now
- **Characteristics:**
  - NLP task are modeled entirely by relying on LMs
  - The tasks of both shallow, deep features extraction and prediction of the data are all given to the LM.
  - Engineering of prompts is required.
- **Representative work:**
  - **GPT3**

What is Prompting?

Encouraging a pre-trained model to make particular predictions by providing a "**prompt**" specifying the task to be done.



General workflow of Prompting

Prompt Addition

Given input x, we transform into prompt x' through **2 steps:**

- Define a template with 2 slots, one for input [x] and one for the answer [z].
- Fill the slot [x]

Example: Sentiment Classification

Input x = "I Love this movie"

↓

Template: [x]Overall, it was a [z] movie

↓

Prompting: X' = " I love this movie. Overall it was a [z] movie

↓

Predicting: x' = "I love this movie. Overall it was a fantastic movie."

Answer Search

- At any stage our LM is doing softmax over the set of vocabulary.
- It might be possible that our model might give more than one word output that might fit the particular sentences.

Example : In case of sentiment Analysis :
z = {excellent,good,ok,bad,horrible}
y = {++,+,~,-,-}
So,

$$\hat{z} = \underset{z \in \mathcal{Z}}{\text{search}}\, P(f_{\text{fill}}(x', z); \theta).$$

So, we can choose a word that maximizes the likelihood of the entire sentences.

Answer-Label Mapping

- Now, we got highest scoring output z.but, for some cases,where multiple answer could result in the same output. So, we can use multiple word to represent single class to hold sentiment.

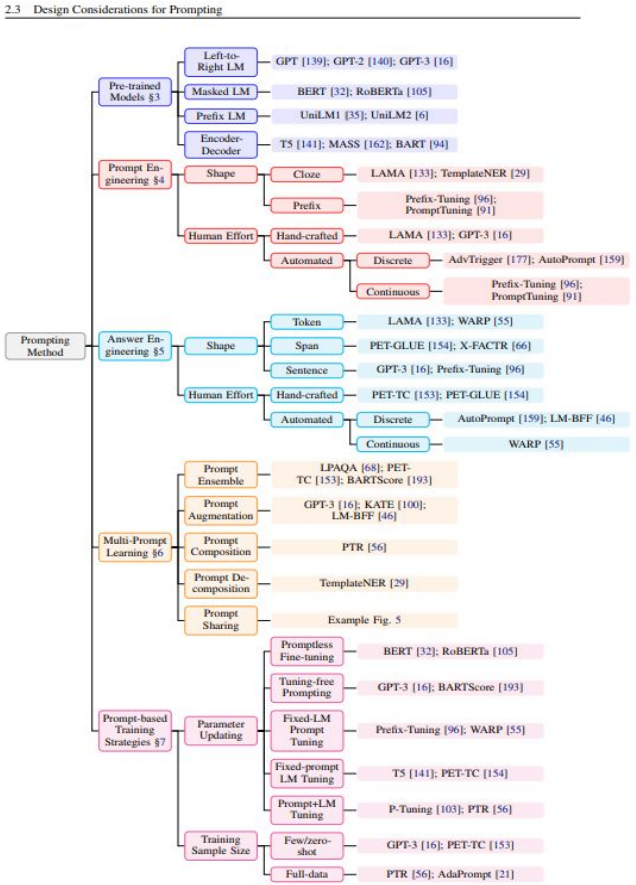
Input ([X])	Template	Answer ([Z])
I love this movie.	[X] The movie is [Z].	great fantastic ...
He prompted the LM.	[X] The text is about [Z].	sports science ...
What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...

Type of prompt

- Prompt : I Love this movie. Overall it was a [z] movie.
- Filled Prompt : I Love this movie. Overall it was a boring movie.
- Answered Prompt: I Love this movie. Overall it was a fantastic movie
- Prefix Prompt : I Love this movie. Overall this movie is [z]
- Cloze Prompt : I Love this movie. Overall it was a [z] movie

Design Considerations for Prompting

- Pre-trained Model
- Prompt Engineering
- Answer Engineering
- Multi-Prompt Learning (Expanding the Paradigm)
- Prompt-based Training Strategies



1. Pre-trained Model

- These are the model which are train specifically to underline the language properties, rule or syntax.

3.1 Training Objective:

- **Standard Language Model(SLM):**
  - Training the model to optimize the probability p(x) of the text from the training corpus. The text generally is in autoregressive fashion.
  - Alternative of standard Lm objective are denoising objective. which we apply some noising function. x' = $f_{noise}(x)$
  - **Corrupted Text Reconstruction(CTR)**:
    - Restore the preprocessed text to its uncorrupted state by calculating loss over only the noise part. For example. Input: I Love this movie. x' = I [z] this movie.
  - **Full Text Reconstruction(FTR):**
    - It calculate the loss over a entirely of the input text. Example: Seq to Seq architecture in and the task is translation.

3.2 Noising Function:

Operation	Element	Original Text	Corrupted Text
Mask	one token	Jane will move to New York .	Jane will [Z] to New York .
	two tokens	Jane will move to New York .	Jane will [Z] [Z] New York .
	one entity	Jane will move to New York .	Jane will move to [Z] .
Replace	one token	Jane will move to New York .	Jane will move [X] New York .
	two tokens	Jane will move to New York .	Jane will move [X] [Y] York .
	one entity	Jane will move to New York .	Jane will move to [X] .
Delete	one token	Jane ~~will~~ move to New York .	Jane move to New York .
	two token	Jane ~~will move~~ to New York .	Jane to New York .
Permute	token	Jane will move to New York .	New York . Jane will move to
Rotate	none	Jane will move to New York .	to New York . Jane will move
Concatenation	two languages	Jane will move to New York .	Jane will move to New York . [/s] 简将搬到纽约。

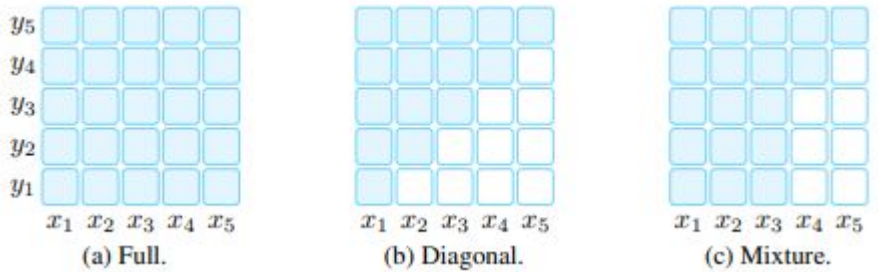
Table 4: Detailed examples for different noising operations.

1. Pre-trained Model

3.3 Directionality of Representation:

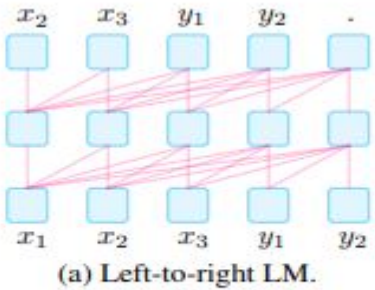
- **Left-to-Right :**
  - The representation of the each word is calculated based on the word itself and all the previous word in the sentence. Example : "This is a good movie" the representation of the word good is would be calculated based on the Previous word. Example GPT
- **Bidirectional:**
  - The representation of the each word is calculated based on the all the word in a sentences. Example : BERT
- **Attention Masking:**
  - We have attention during the training that calculate the weight we want to put to neighbour words while making a prediction on certain time 't'.
  - It helps to select the which neighbour we are talking about

- **For full attention**, for any 'y' that you generate you have to put your attention to all the words
- **For Diagonal**, for any 'y' that you want to generate put the attention to all the previous generated word.
- **For Mixture attention**, you will always have to give the attention of the all previous word as well as some prefix token.



1. Pre-trained Model

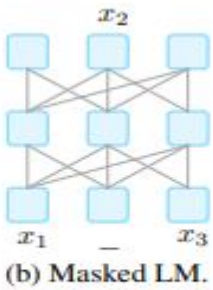
Left-to-right LM



(a) Left-to-right LM.

- For any time 't' we have to see all the words that already occured.
- Example: GPT-1,GPT-2,GPT-3

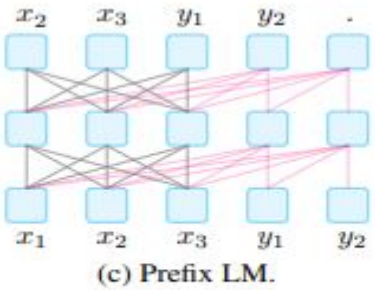
Masked LM



(b) Masked LM.

- If we want to predict any word, we should make the attention to all the words we have in the sentence.
- Example: BERT

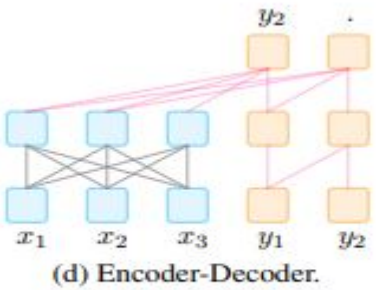
Prefix LM



(c) Prefix LM.

- It follows full attention mask and starts predict next word and they will also put the attention on the generated word.
- Examples: UniLm 1,2, ERNIE-M

Encoder-Decoder

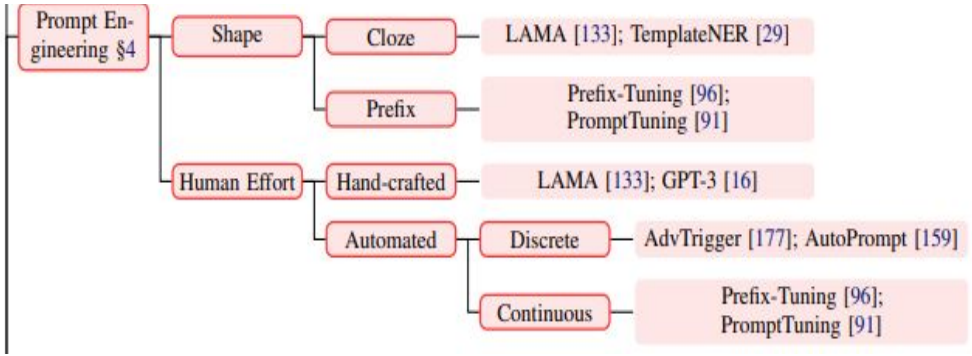


(d) Encoder-Decoder.

Example : BART, T5

2. Prompt Engineering

- Process of creating a prompt function $f_{prompt}(x)$ in which we try to fit into the input sentences and place a mask token for model prediction.



Prompt Shape: How to define the shape of a prompt template?

There are 2 varieties:

- **Cloze prompts:**
  - prompt with a slot [z] to fill in the middle of the text as a cloze prompt. can be used in task such as fill in the blanks or text classification.
  - **Example**: I love this movie. Overall it was a [z] movie.

- **Prefix prompts:**
  - prompt where the input text comes entirely before slot [z]. can be used in text generation.
  - **Example**: I love this movie. Overall this movie is [z].

2. Prompt Engineering

Design of Prompt Templates:

- **Hand-Crafted:**
  - configure the manual template based on the characteristics of the task.
  - Required a group of people.
  - It is not scalable and also so many pattern are not possible by human

- **Automated Search:**
  - 2 methods
    - Discrete prompt
    - Continuous prompt

Representative Method for Discrete prompt:

- **Prompt Mining**
  - Mine prompts given a set of questions/answers**.**
  - **Middle-words.**
    - **Example**: Barack Obama was born in USA. —> [X] was born in [y]
  - **Dependency-based**
    - **Example** : The capital of France is Paris → capital of [x] is [y]

- **Prompt Paraphrasing**
  - Paraphrase the existing prompt and get new prompt
  - back translation with beam search.
  - **Example:** [X] shares a border with [Y] —>Encoder-decoder→[X] has common border with [Y]
    —> [X] adjoins [Y]

2. Prompt Engineering

- **Gradient-based Search:**
    - Applied a gradient based search over a actual tokens to find short sequences that can trigger the pretrained LM model to generate the desired target prediction.
    - To form initial sequence they start concatenating the stop words, keep replacing these word such that the overall loss decrease and probability of target prediction increases
    - for classification task target word can be 1 or LM it can be 2 or 4 words

- **Prompt Generation:**
    - It is a text generation task where they used T5 model for generation task.
    - For T5 model we have a prefix token and we have {1…n} words we randomly mask the sequence of the words all these mask words and prefix token is given to the Encoder and Decoder model and it predict the mask token. All it happens from text to text
    - So, they use T5 to generate a template token
        - Specifying the position to insert template tokens within a template
        - provide training samples for T5 to generate to decode template token

- **Prompt Scoring**
    - To investigate the task of knowledge base completion and design a template for an input(head-relation-tail triple) using LM.
    - They first hand-craft a set of templates as a potential candidates and fill the input and answer slots to form a filled prompts.

2. Prompt Engineering

Representative Method for Continuous prompt:

- Continuous prompts remove the constraint of using natural language as the input. This means that the prompt can be in any form that can be represented in the embedding space of the language model. This allows for more flexibility in the way prompts are constructed and can potentially lead to better performance on specific tasks.
- Continuous prompts have their own parameters that can be tuned based on training data from the downstream task. This means that the prompt construction is not limited to the pre-trained parameters of the language model, but can be customized to better fit the specific task at hand.

Prefix Tuning :

- Prefix Tuning is a method that adds a sequence of continuous task-specific vectors to the input. This is done while keeping the parameters of the pre-trained language model frozen.

$$\max_{\phi} \log P(\boldsymbol{y}|\boldsymbol{x};\theta;\phi) = \max_{\phi} \sum_{y_i} \log P(y_i|h_{<i};\theta;\phi).$$

- The method involves optimizing over a log-likelihood objective function that maximizes the probability of generating the output sequence given the input and the prefix. This is done by tuning a trainable prefix matrix and a fixed pre-trained LM parameterized by θ.

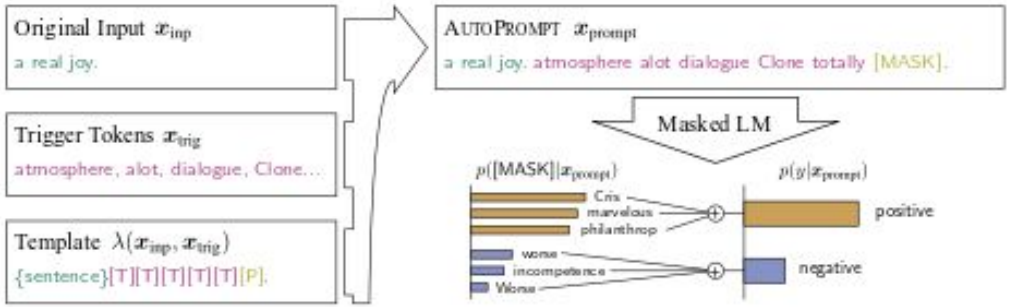
2. Prompt Engineering

Tuning Initialized with discrete Prompts:

- This method involves using a **discrete prompt search** method (AutoPrompt) to discover a template, which is then used to initialize virtual tokens for a continuous prompt search. The embeddings of these virtual tokens are fine-tuned to increase task accuracy.

- The initial set of templates used in this method can either be manually crafted or obtained using prompt mining methods. Studies have found that initializing with manual templates can provide a better starting point for the search process, and that jointly learning weights and parameters for each template can further improve performance.

- **AutoPrompt** is a discrete prompt search method that generates effective prompts for language models by iteratively adding tokens to a template and evaluating the resulting prompt on a downstream task.

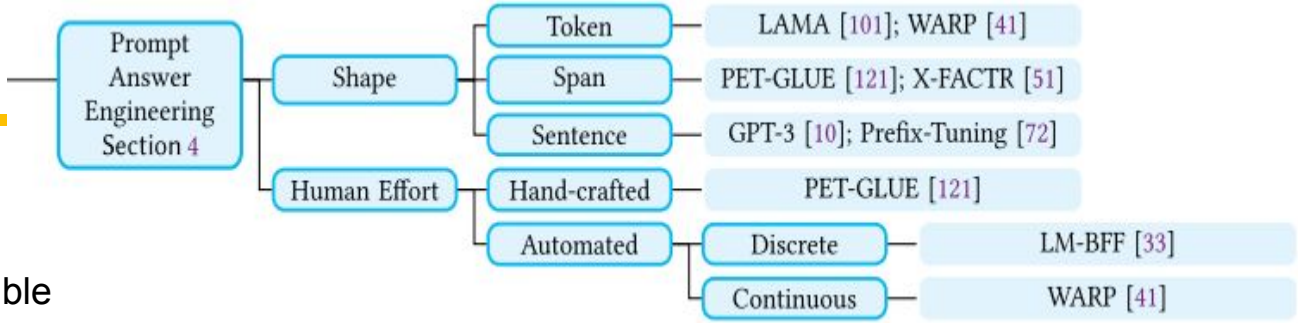


Hard-Soft Prompt Hybrid Tuning

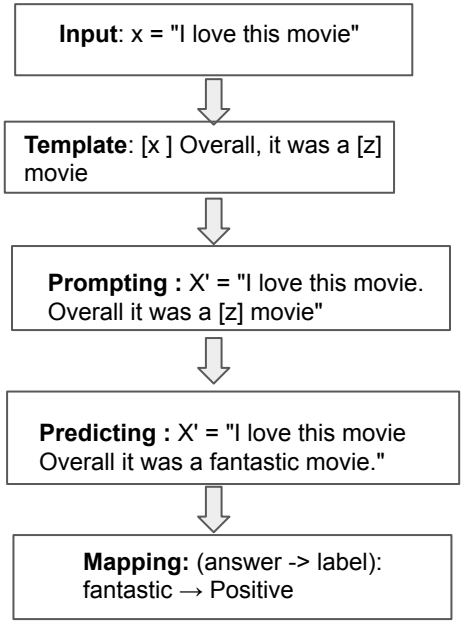
- Prompt tuning methods improve the effectiveness of prompt templates by combining learnable embeddings with a hard prompt template.

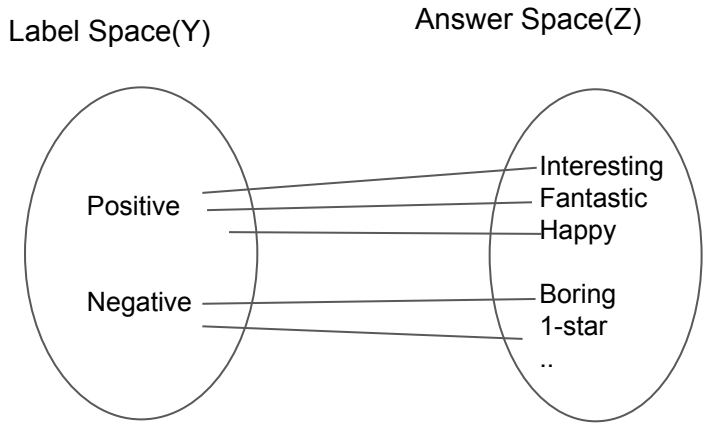
- Liu et al.'s "P-tuning" method uses continuous prompts learned by inserting trainable variables into embedded input, while Han et al.'s "PTR" method uses manually crafted sub-templates composed with logic rules and virtual tokens whose embeddings can be tuned with pre-trained LMs parameters. Both methods introduce **task-related anchor tokens** to further improve the prompt template. Experimental results demonstrate the effectiveness of these prompt tuning methods in relation classification tasks.

3. Answer Engineering



- Answer Engineering is the engineering the set of possible outputs
- It aims to search for an answer space and a map to the original output y that results in an effective predictive model.
- **Shape** : How to define the shape of an answer?
- **Human Effort** : How to search for appropriate answers?





3. Answer Engineering

Answer Shape:

- **Token :** Answers can be one or more tokens in the pretrained language model vocabulary.
- **Span:** Answers can be chunks of words made up of more than one tokens. These are usually used together with cloze prompts
- **Sentence**: Answers can be a sentences of arbitrary length. Usually used with prefix prompt

Type	Task	Input ([X])	Template	Answer ([Z])
Text CLS	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1] [X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...

Token or Span

Sentence

Human Effort

Hand-crafted:

- **Unconstrained answer space:**
  - It is common to directly map the answer z to the final output y using the identity mapping. This approach simplifies the process of generating an answer by avoiding the need for complex transformations or additional processing steps.

- **Constrained answer space :**
  - In some natural language processing tasks, the space of possible outputs is constrained, such as in text classification or entity recognition tasks, where the output is limited to a predefined set of labels or categories. This allows for easier processing and analysis of the output data, as the range of possible outputs is limited and well-defined.
  - To facilitate the mapping between the answer space Z and the underlying class Y in constrained spaces, manual design of lists of relevant words or categories is often performed. For example, for entity recognition tasks, lists of relevant entity types such as "person", "location", etc. are manually created. In multiple-choice question answering, a language model can be used to calculate the probability of an output among multiple choices.

3. Answer Engineering

Automated Search :

Discrete Search Space :

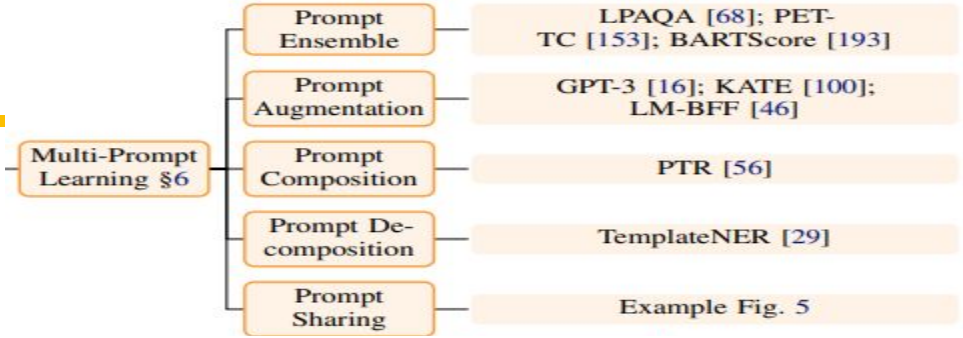
- **Answer Paraphrasing**
  - To perform answer paraphrasing, Lets say, a pair of answer and output (z' , y). We define a function that generates a paraphrased set of answers para(z').The probability of the final output is then calculated as the marginal probability of all answers in the paraphrase set.
  - **Example**: Initial Answer (z'): The capital of France is Paris.
  - Paraphrased Answer (y) : Paris is the capital of France, France's capital city is Paris, In France, the capital is Paris.
  - answer paraphrasing is used to generate multiple variations of the initial answer, which can improve the language model's accuracy and coverage of the answer space. These paraphrased answers convey the same information as the initial answer but in different ways, making it easier for the model to capture the correct answer.

- **Prune-then-Search**
  - A initial pruned answer space of several plausible answers is generated
  - An algorithm further searches over this pruned space to select a final set of answers

- **Label Decomposition**
  - decompose each relation label into its constituent words and use them as a answer.
  - city_of_death -> {person,city,death}

Continuous Search Space

- Assign a virtual token for each class label and optimize the token embedding for each label.



Prompt Ensembling

Definition:

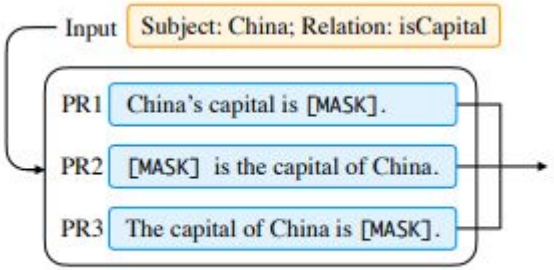
- Using multiple unanswered prompts for an input at inference time to make predictions

Advantages:

- Utilize complementary advantages
- stabilize performance on downstream task

Methods:

- Uniform Averaging
- Weighted Averaging
- Majority Voting



(a) Prompt Ensembling.

4. Multi- Prompt Learning

Prompt Augmentation:

- Prompt augmentation, also known as demonstration learning, involves providing a few additional answered prompts to the LM to demonstrate how it should provide an answer to the actual prompt.

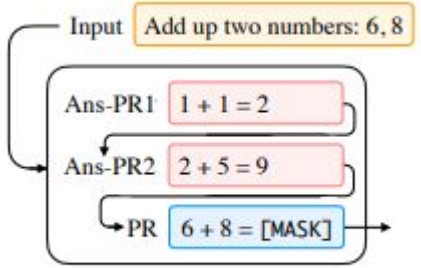
Prompt Composition:

- Prompt composition involves using multiple sub-prompts to break down a complex task into more fundamental subtasks, and then defining a composite prompt based on those sub-prompts.
- This technique is particularly useful for composable tasks, such as relation extraction, where the task can be broken down into several subtasks, each with its own sub-prompt.

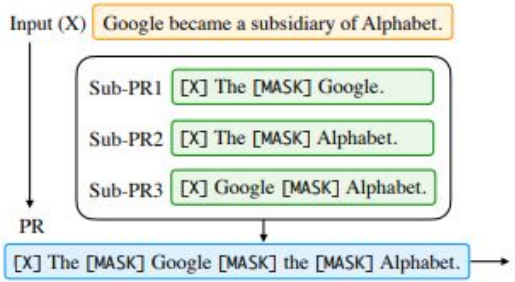
Prompt Decomposition:

- Prompt decomposition is a technique used to address the challenge of defining a holistic prompt for tasks where multiple predictions should be performed for one sample, such as sequence labeling tasks like named entity recognition.

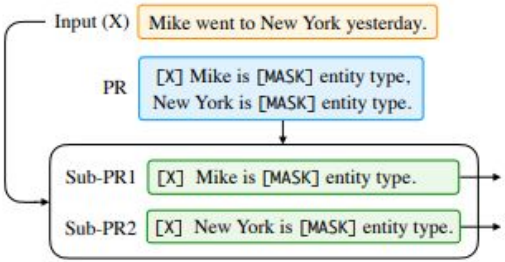
- In prompt decomposition, the holistic prompt is broken down into different sub-prompts, with each sub-prompt used to prompt the model to make a prediction for a specific aspect of the input, such as predicting the entity type for each text span in a sentence for named entity recognition. This allows for more accurate and efficient predictions by handling each sub-prompt separately.



(b) Prompt Augmentation.

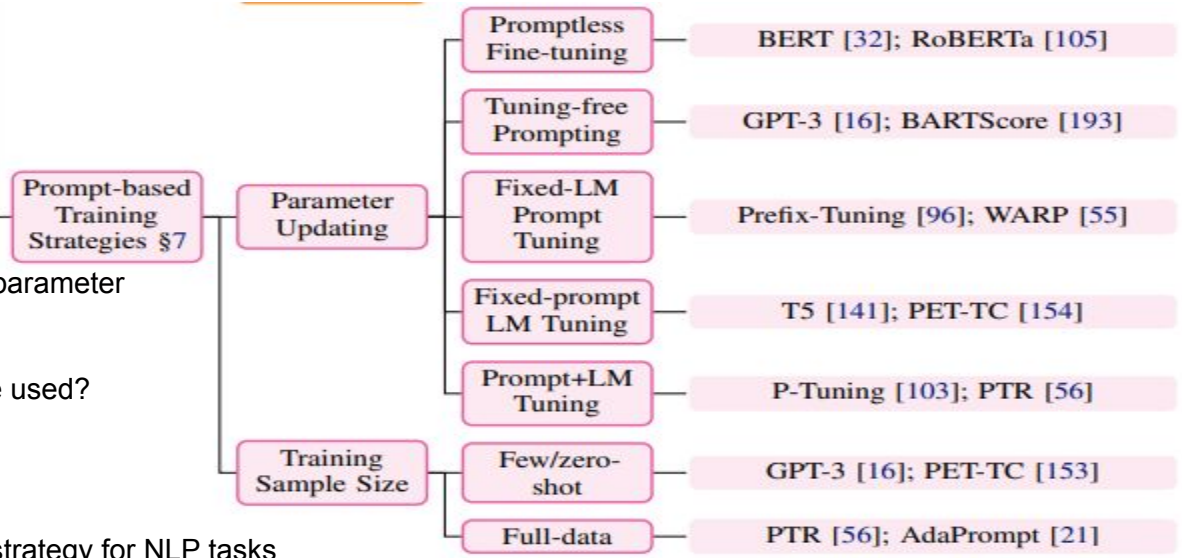


(c) Prompt Composition.



(d) Prompt Decomposition.

5. Prompt-based Training Strategies:



Parameter Perspective: Explain about how are parameter updated?

Training sample: How many training samples are used?

Promptless Fine-tuning:

- Promptless fine-tuning is a widely-used strategy for NLP tasks that involves fine-tuning a pre-trained language model on a downstream task without the use of prompts. This strategy involves updating all or some of the parameters of the pre-trained LM based on gradients induced from the training data of the downstream task.
- While promptless fine-tuning is a simple and powerful method, it may suffer from overfitting or unstable learning on small datasets. Additionally, models trained via promptless fine-tuning may be prone to catastrophic forgetting.

5. Prompt-based Training Strategies:

Tuning-free Prompting:

- One advantage of tuning-free prompting is that it can generate answers more efficiently than fine-tuning, as it does not require updating the pre-trained LM parameters. This makes it particularly useful for generating answers on the fly, such as in chatbot applications or other real-time applications.
- Tuning-free prompting can also be more robust than fine-tuning, as it does not risk overfitting or catastrophic forgetting. However, it may be less accurate than fine-tuning in some cases, as it relies solely on the pre-trained knowledge of the LM and cannot be optimized for specific tasks.

Fixed-LM Prompt Tuning

- Fixed LM prompt tuning is a prompting method that only updates the parameters of the prompt, while keeping the parameters of the pre-trained language model unchanged. This approach is used in Prefix-Tuning and WARP, where the prompts' parameters are updated based on the supervision signal obtained from the downstream training samples.
- The advantage of fixed LM prompt tuning is that it is computationally efficient and requires much less fine-tuning time than traditional fine-tuning methods

Fixed-prompt LM Tuning:

- Fixed-prompt LM tuning involves tuning the parameters of the pre-trained LM along with prompts that have fixed parameters to specify the model behavior. This approach can potentially lead to improved performance, especially in few-shot scenarios.
- PET-TC, PET-Gen, and LM-BFF are examples of fixed-prompt LM tuning methods that use discrete textual templates applied to every training and test example to guide the model's behavior.

Applications:

Knowledge Probing:

Factual probing:

- Factual probing is used to evaluate the amount of factual knowledge present in the pre-trained LM's internal representations, and is one of the earliest scenarios for prompting methods.
- In factual probing, pre-trained model parameters are typically fixed, and knowledge is retrieved by transforming the input into a cloze prompt, either manually or automatically. Relevant datasets include LAMA and X-FACTR. Both discrete and continuous template learning, as well as prompt ensemble learning, have been explored in this context.

Linguistic Probing:

- Linguistic probing tasks aim to evaluate the linguistic capabilities of pre-trained language models by presenting them with natural language sentences to complete. These tasks can assess various linguistic phenomena such as analogies, negations, semantic role sensitivity, semantic similarity, and rare word understanding.
- Different prompting methods have been applied to linguistic probing tasks, including discrete template search, continuous template learning, and prompt ensemble learning. These methods can be used to generate effective prompts that can improve the performance of language models on linguistic probing tasks.

Classification-based Tasks

Text-classification Tasks:

- The use of prompt learning has been explored extensively for text classification tasks in the context of few-shot settings with "fixed-prompt LM Tuning" strategies.

Challenges

1. **Prompt Design :**
   - **Tasks beyond Classification and Generation:**
     The design of prompts for tasks beyond classification and generation is less straightforward, which has limited the application of prompt-based learning to information extraction and text analysis tasks. To effectively apply prompting methods to these tasks, researchers may need to reformulate the tasks so that they can be solved using classification or text generation-based methods or use effective answer engineering techniques to express structured outputs in a suitable 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.
   - **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.

2. Answer Engineering:
   - **Many-class and Long- answer Classification Tasks:**
     - When there are too many classes, how to select an appropriate answer space becomes a difficult combinatorial optimization problem
     - When using multi-token answers, how to best decode multiple tokens using LMs remains
   - **Multiple Answers for Generation Tasks**
     - For text generation tasks, qualified answers can be semantically equivalent but syntactically diverse
     - How to better guide the learning process with multiple references remains a largely open research problem

3. 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
   - **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.

Thank you for your time 😊