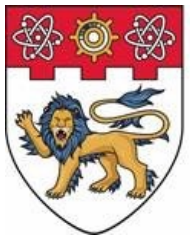


NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

An Introduction to Prompting Methods

Yige Xu

PhD Student, SCSE, NTU

yige002@e.ntu.edu.sg

https://xuyige.github.io

Outlines

- Development Line before Prompting
- Formulation of Prompting
- Prompt Learning and Training Strategies
- Verbalizer Learning
- Prompting Methods
- Recent Applications on Prompt

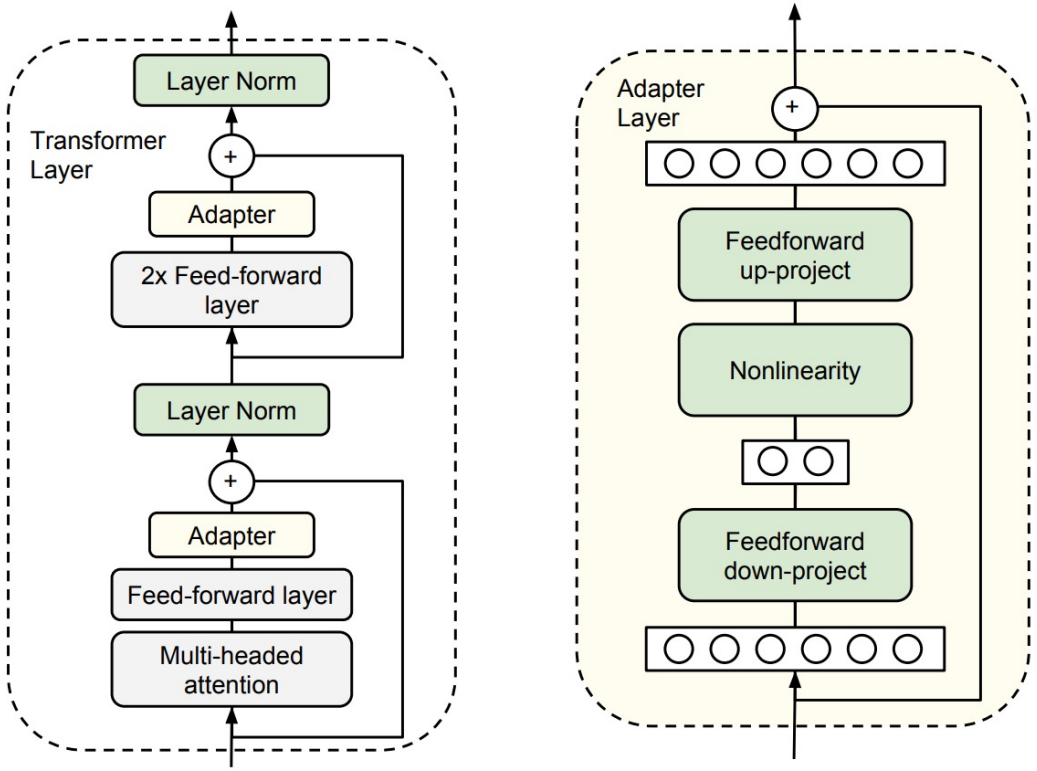
Development Line before Prompting

- Pre-trained Language Models
  - Pre-train in large-scale unlabeled corpus and then fine-tune in “small-scale” downstream data have brought breakthrough on many NLP tasks
  - It is hard to tune all parameters when PLM is becoming larger and larger

- Two solutions
  - Model compression and inference speedup, including knowledge distillation, quantization, parameter sharing, module replacing, and early exit
  - Parameter-efficient tuning methods, including Adapters, Part Parameter Tuning, and Prompt Tuning

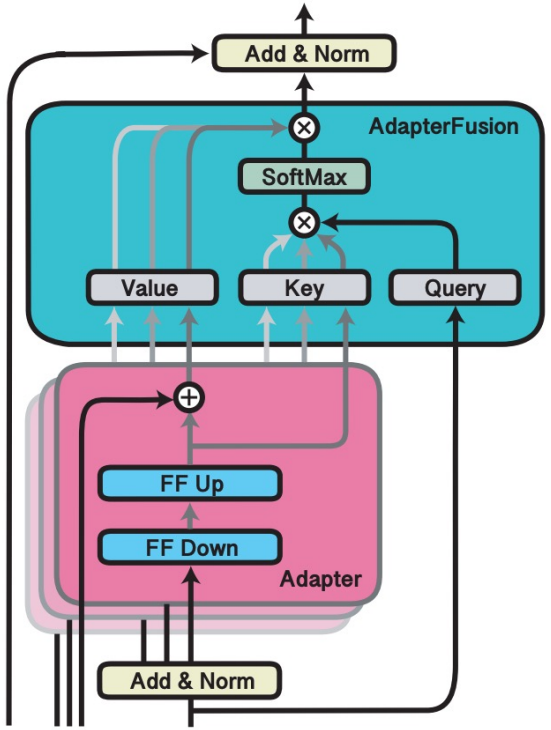
Development Line before Prompting

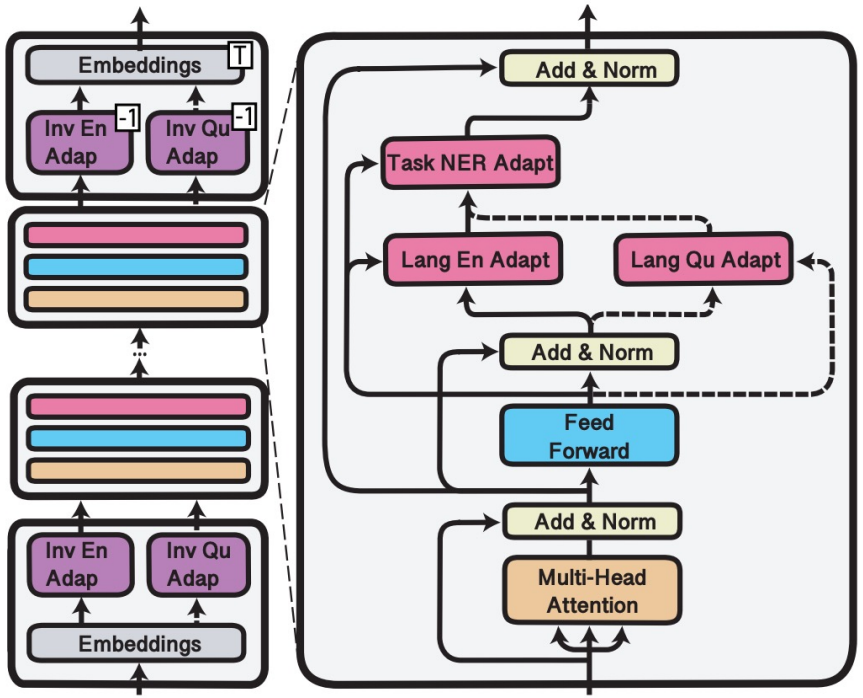
- Adapters
  - Add the adapter module inside the Transformer layer and fix the PLM

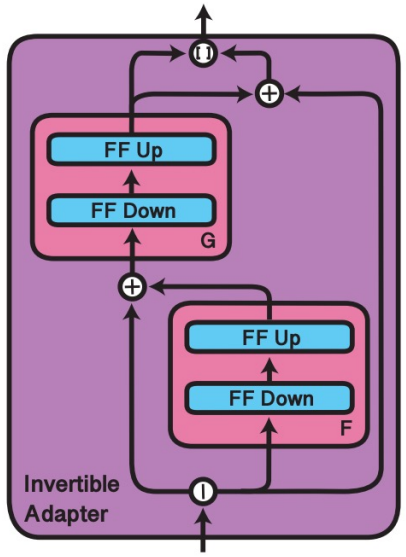


Development Line before Prompting

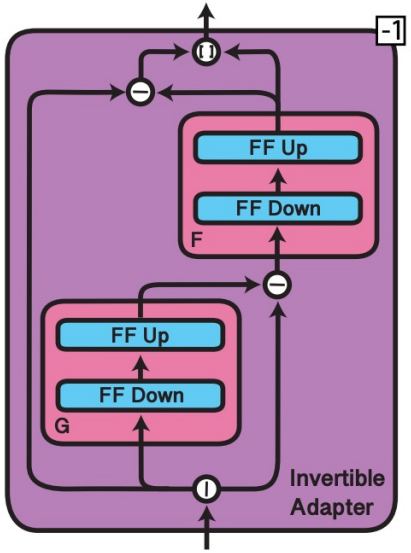
- Adapters
  - Can be used in Cross-task/Cross-lingual/Cross-Modal transfer







(a) The invertible adapter



(b) The inversed adapter

Development Line before Prompting

- Part Parameter Tuning
  - *BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models* (ACL 2022) Only the bias-term (or a subset of them) is fine-tuned
  - *Parameter-Efficient Transfer Learning with Diff Pruning* (ACL 2021) Learn a diff vector and add this vector to the pre-trained model parameters
  - *Raise a Child in Large Language Model: Towards Effective and Generalizable Fine-tuning* (EMNLP 2022) First find a subset of parameters and generate the gradient mask, then update part of the parameters based on the gradient mask

Formulation of Prompting

- Pattern-Exploiting Training
  - M: Masked Language Model
  - L: A Set of Labels
  - V: Vocabulary
  - x: Input Sequence
  - [MASK]: Masked Token, in V
  - pattern: A function P that takes x as input and outputs a phrase or sentence P(x) in V*
  - verbalizer: An injective function v: L->V
  - Pattern-verbalizer pair (PVP): (P, v)

Formulation of Prompting

- Notations

Name	Notation	Example	Description
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	`[X]` Overall, it was a `[Z]` movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot `[Z]` where answer $\boldsymbol{z}$ may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a `[Z]` movie.	A text where `[X]` is instantiated by input $\boldsymbol{x}$ but answer slot `[Z]` is not.
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot `[Z]` is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot `[Z]` is filled with a true answer.
Answer	$\boldsymbol{z}$	"good", "fantastic", "boring"	A token, phrase, or sentence that fills `[Z]`

Prompt Learning and Training Strategies

- Prompt Engineering
  - Prompt engineering is the process to create a prompting function $f_{\text{prompt}}(x)$ that maximize the performance on downstream tasks.
  - In many previous works, one or more templates are searched for each task.

- Prompt Types:
  - Cloze Prompt: The language model is required to fill in the blanks of a textual string.
  - Prefix Prompt: The language model is required to generate the textual string by the given input as well as a prefix string.

Prompt Learning and Training Strategies

- Manual Template
  - Manually define some patterns:

- p. Question: q? Answer: __.
- p. Based on the previous passage, q? __.
- Based on the following passage, q? __. p

$P_1(\mathbf{x}) =$ ___: a b $P_2(\mathbf{x}) =$ a (___) b

$P_3(\mathbf{x}) =$ ___ – a b $P_4(\mathbf{x}) =$ a b (___)

$P_5(\mathbf{x}) =$ ___ News: a b

$P_6(\mathbf{x}) =$ [Category: ___] a b

$P_1(a) =$ It was ____. a $P_2(a) =$ Just ____! || a

$P_3(a) =$ a. All in all, it was ____.

$P_4(a) =$ a || In summary, the restaurant is ____.

  - Advantages:
    - Help leverage the knowledge contained in pre-trained language model
    - Can be manual designed with prior expertise
    - Provide a perspective of better understanding the pre-trained language model
  - Disadvantages:
    - Not flexible enough
    - Require much prior expertise

Prompt Learning and Training Strategies

- Automated Template
  - Automatically search templates in discrete space (word index) or continuous space (word embedding)
  - Methods:
    - Discrete Prompts, include Prompt Mining (Jiang et. al. 2020), Prompt Paraphrasing (Yuan et. al. 2021, Haviv et. al. 2021), Gradient based Search (Shin et. al. 2020), etc.
    - Continuous Prompts. Include Prefix Tuning (Li and Liang, 2021), Tuning Initialized with Discrete Prompts (Zhong et. al. 2021, Shin et. al. 2020), Hard-Soft Prompt Hybrid Tuning (Liu et. al. 2021, Han et. al. 2021), etc.
  - Advantages:
    - Relax the constraint of embedding space
    - Remove the restriction that the template is parameterized by the PLM parameters
  - Disadvantages:
    - Can hardly be interpretable
    - Usually, cannot be easily transferred

Prompt Learning and Training Strategies

- Training Settings
  - Zero-shot learning (PLM has been trained for LM to predict the probability, GPT-3)
  - Few-shot learning (Resource limitation scenarios, PET-TC)
  - Full-data learning (For best downstream task performance, PTR)
- Parameter Updating
  - No prompt, only fine-tuning: Only tuned LM params (BERT, RoBERTa)
  - Tuning-free prompting: Frozen LM params, no additional prompt params (GPT-3, BARTScore)
  - Fixed-LM prompt tuning: Frozen LM params, additional and tuned prompt params (Prefix-Tuning, WARP)
  - Fixed-prompt LM Tuning: Tuned LM params, with fixed additional prompt params (T5, PET-TC)
  - Prompt + LM tuning: Tune both LM params and additional prompt params (P-Tuning, PTR)

Prompt Learning and Training Strategies

- Parameter Updating
  - No prompt, only fine-tuning: Only tuned LM params
    - Advantages: Simplicity, do not need prompt design. Can also fit to larger training datasets.
    - Disadvantages: LM usually overfit or unstable on small datasets.
    - Examples: BERT, RoBERTa
  - Tuning-free prompting: Frozen LM params, no additional prompt params
    - Also referred as “in-context learning”: [TASK DESCRIPTION] x1 [SEP] y1 [SEP] x2 [SEP] y2 [SEP] x [SEP] [MASK] ?
    - Advantages: No parameter update process, which can be applied in zero-shot settings.
    - Disadvantages: (1) Requires much prompt engineering with prior expertise; (2) Slow at testing, cannot easily use large training datasets.
    - Examples: GPT-3, BARTScore

Prompt Learning and Training Strategies

- Parameter Updating
  - Fixed-LM prompt tuning: Frozen LM params, additional and tuned prompt params
    - Advantages: Often outperforms tuning-free prompting, while retain knowledge in LMs and is suitable in few-shot scenarios (similar to tuning-free prompting).
    - Disadvantages: (1) Not applicable in zero-shot scenarios; (2) The learned representation has limitations in large-data scenarios (though effective in few-shot scenarios); (3) Prompts are usually hard to human-interpreted.
    - Examples: Prefix-Tuning, WARP
  - Fixed-prompt LM Tuning: Tuned LM params, with fixed additional prompt params
    - Advantages: Prompt or answer engineering is more specific to the task, allowing for more efficient learning.
    - Disadvantages: (1) Still requires manual prompt or answer engineering; (2) LM fine-tuned on one downstream task cannot be easily transferred to another downstream task.
    - Examples: T5, PET-TC
    - With null prompt, where directly concatenates “[X] [Z]” without any template words can sometimes achieves competitive accuracy.

Prompt Learning and Training Strategies

- Parameter Updating
  - Prompt + LM tuning:
    - Tune both LM params and additional prompt params.
    - This settings is very similar to the standard pre-train and then fine-tune paradigm.
    - Advantages: Most expressive method, which is suitable for large dataset settings.
    - Disadvantages: Requires most computational resources. Have the similar limitations with pre-train + fine-tune paradigm (e.g., may overfit to small datasets)
    - Examples: P-Tuning, PTR

Verbalizer Learning

- Human-written Verbalizer
  - Manually define some verbalizers (Example: Yelp-5)

$$v(1) = \text{terrible} \quad v(2) = \text{bad} \quad v(3) = \text{okay}$$
$$v(4) = \text{good} \quad v(5) = \text{great}$$

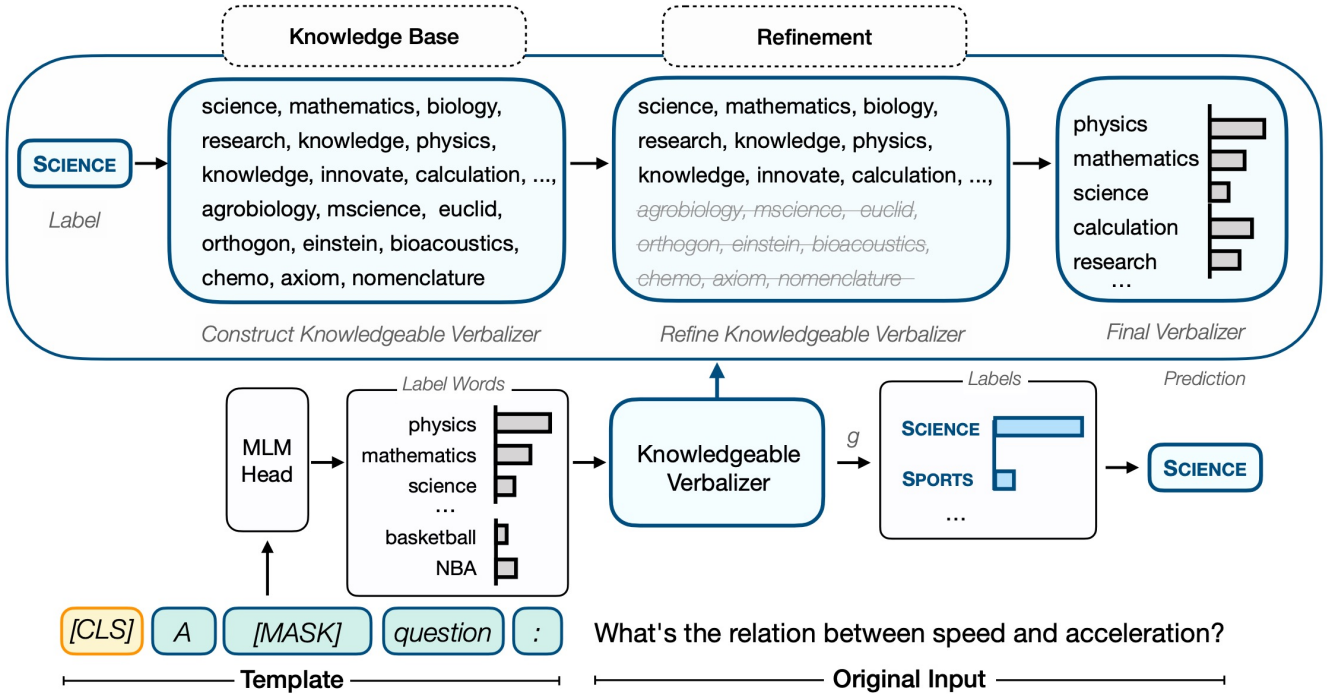
- Automatic Verbalizer:
  - Use multiple verbalizers with a weight

$$q_{\mathbf{p}}(y \mid \mathbf{x}) = \frac{\exp\left(\frac{1}{|v_y|}\sum_{t \in v_y} M(t \mid P(\mathbf{x}))\right)}{\sum_{i=1}^{k} \exp\left(\frac{1}{|v_i|}\sum_{t \in v_i} M(t \mid P(\mathbf{x}))\right)}$$

  - Automating Label Token Selection
    - Train a logistic classifier to predict the class label using the contextualized embedding of the [MASK] token
    - Substitute the hidden state with the MLM's output word embeddings to obtain a score
    - Construct the sets from the k-highest scoring words

Verbalizer Learning

- Knowledgeable Prompt-Tuning
  - Construct external knowledge base by Related Words, ConceptNet, WordNet, etc.
  - Remove the rare words



Prompting Methods

- Theoretical Support
  - *Can Unconditional Language Models Recover Arbitrary Sentences?* (NeurIPS 2019)
  - A PLM can generate a sentence through the identification of a point in the sentence space

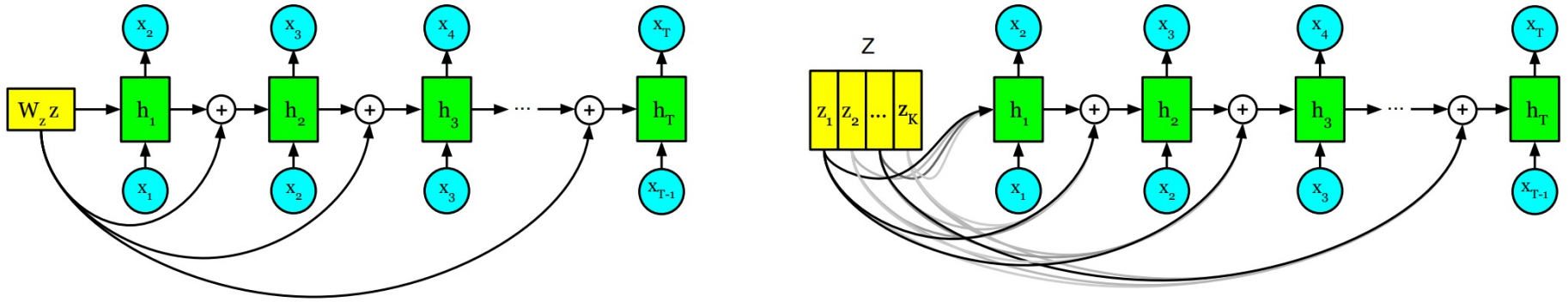
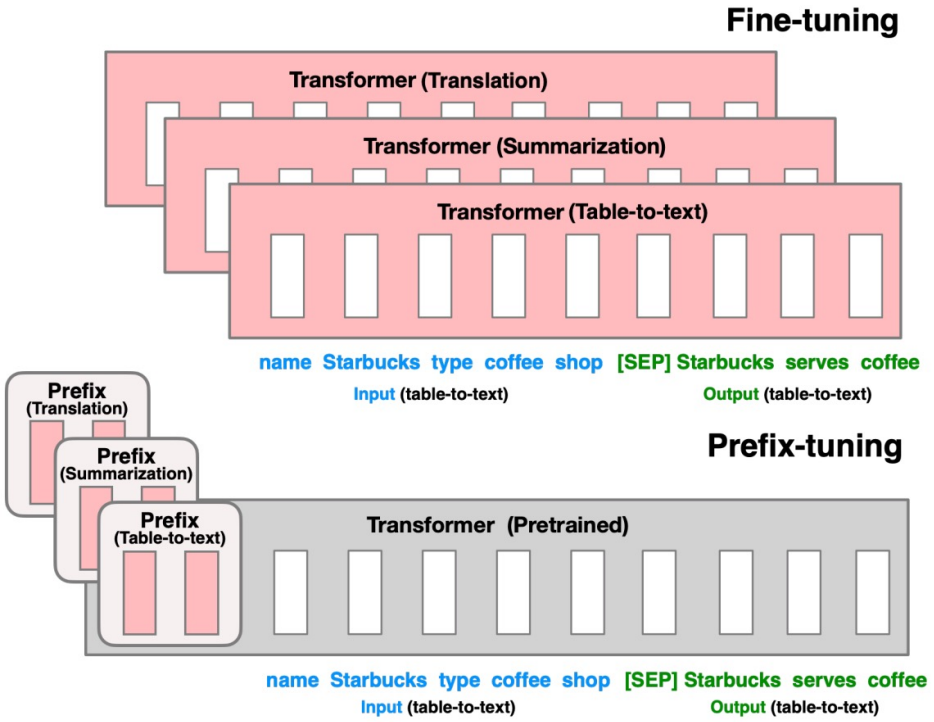


Figure 1: We add an additional bias, $W_z z$ (left, when $\dim(z) \leq d^*$) or $Z = [z_1 \dots z_K]$ (right, when $dim(z) > d^*$), to the previous hidden and cell state at every time step. Only the z vector or Z matrix is trained during forward estimation: The main LSTM parameters are frozen and W_z is set randomly. In the right hand case, we use soft attention to allow the model to use different slices of Z each step.

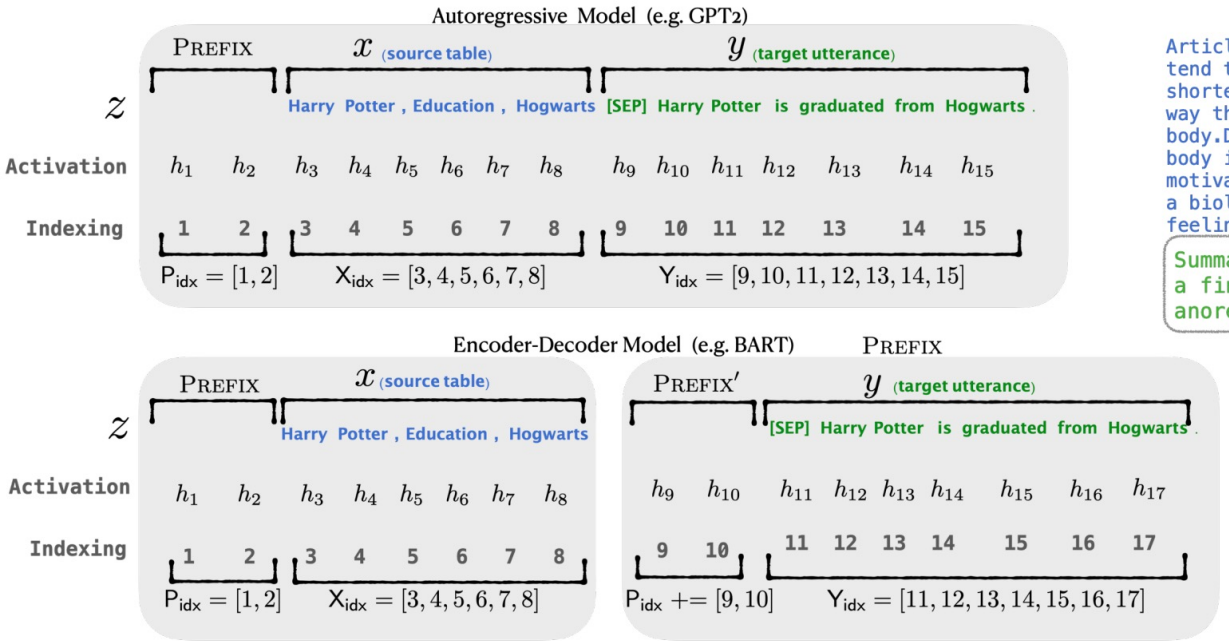
Prompting Methods

- Prefix-Tuning
  - *Prefix-Tuning: Optimizing Continuous Prompts for Generation* (ACL 2021)
  - Add a prefix before the input and only tune the parameters from the prefix



Prompting Methods

- Prefix-Tuning
  - *Prefix-Tuning: Optimizing Continuous Prompts for Generation* (ACL 2021)
  - Add a prefix before the input and only tune the parameters from the prefix



Summarization Example

Article: Scientists at University College London discovered people tend to think that their hands are wider and their fingers are shorter than they truly are.They say the confusion may lie in the way the brain receives information from different parts of the body.Distorted perception may dominate in some people, leading to body image problems ... [ignoring 308 words] could be very motivating for people with eating disorders to know that there was a biological explanation for their experiences, rather than feeling it was their fault."

Summary: The brain naturally distorts body image - a finding which could explain eating disorders like anorexia, say experts.

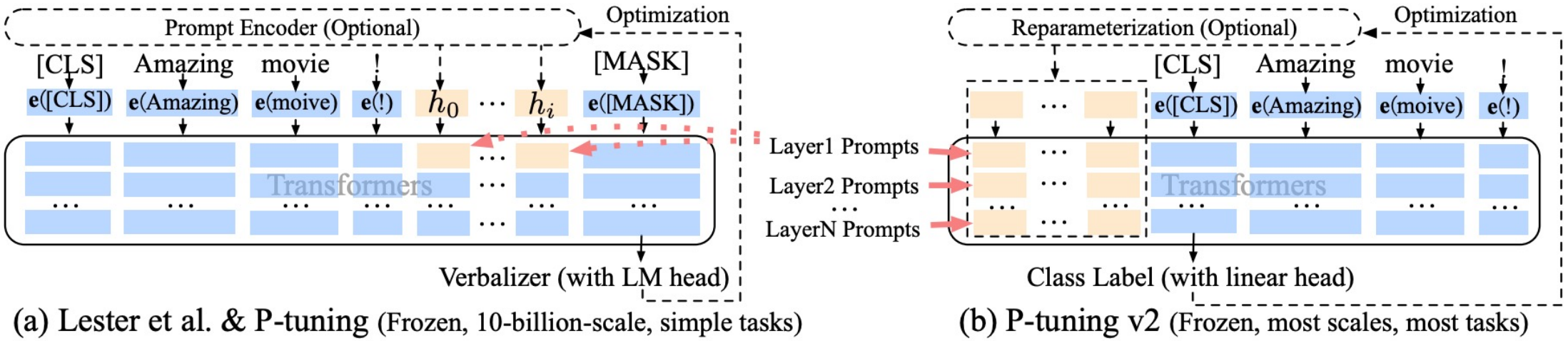
Table-to-text Example

Table: name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

Textual Description: Clowns is a coffee shop in the riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

Prompting Methods

- P-Tuning v2
  - *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks* (ACL 2022)
  - Add prompts in each Transformer layer, which makes the model more stable

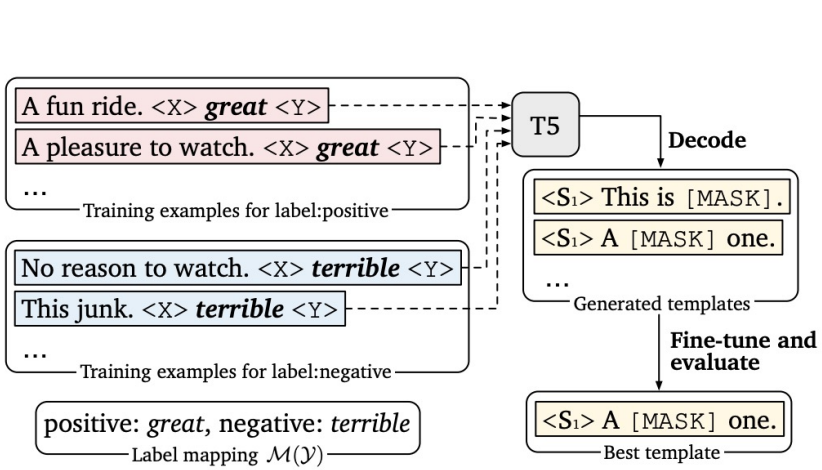


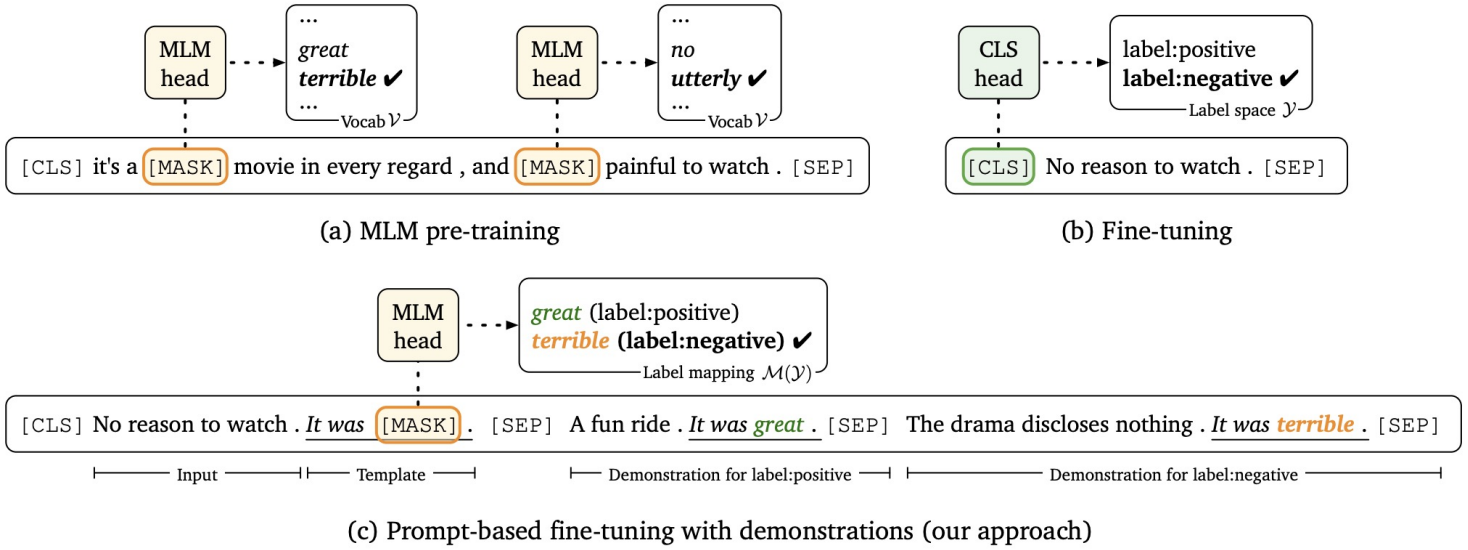
(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

Prompting Methods

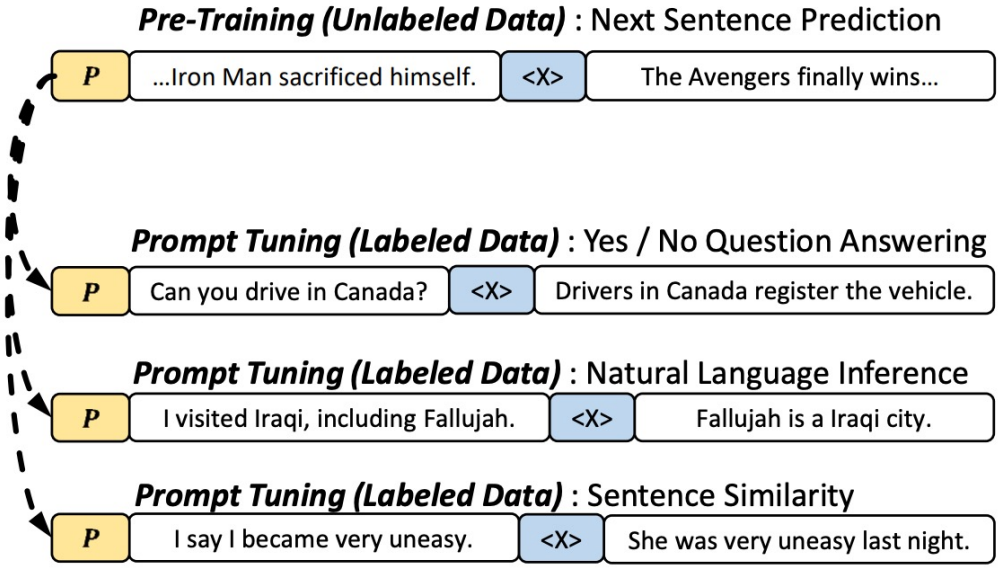
- LM-BFF
  - *Making Pre-trained Language Models Better Few-shot Learners* (ACL 2021)
  - Use T5 to generate the template automatically
  - Add some demonstrations for better comparisons





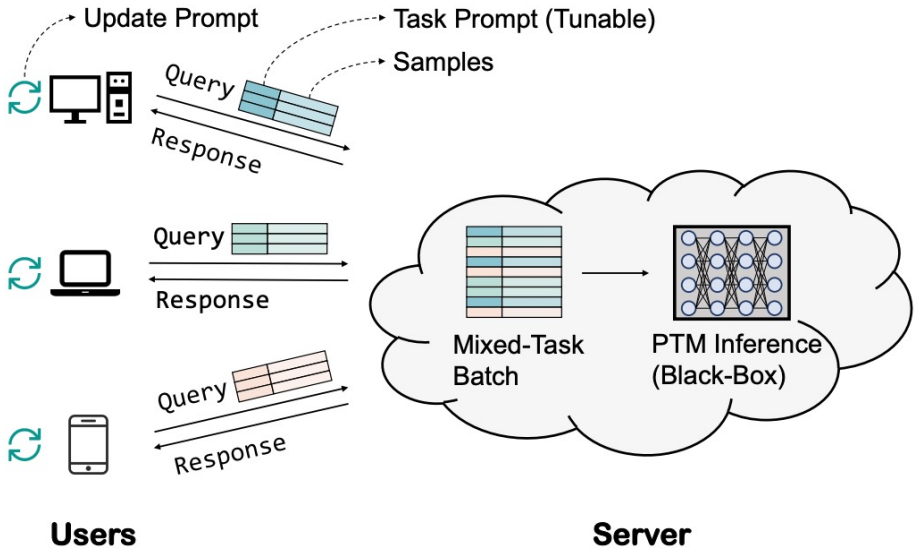
Prompting Methods

- Pre-trained Prompt Tuning
  - *PPT: Pre-trained Prompt Tuning for Few-shot Learning* (ACL 2022)
  - Use three different tasks for prompt pre-training: sentence-pair classification, multiple-choice classification, and single-sentence classification
  - Can obtain a good initialization point of prompts



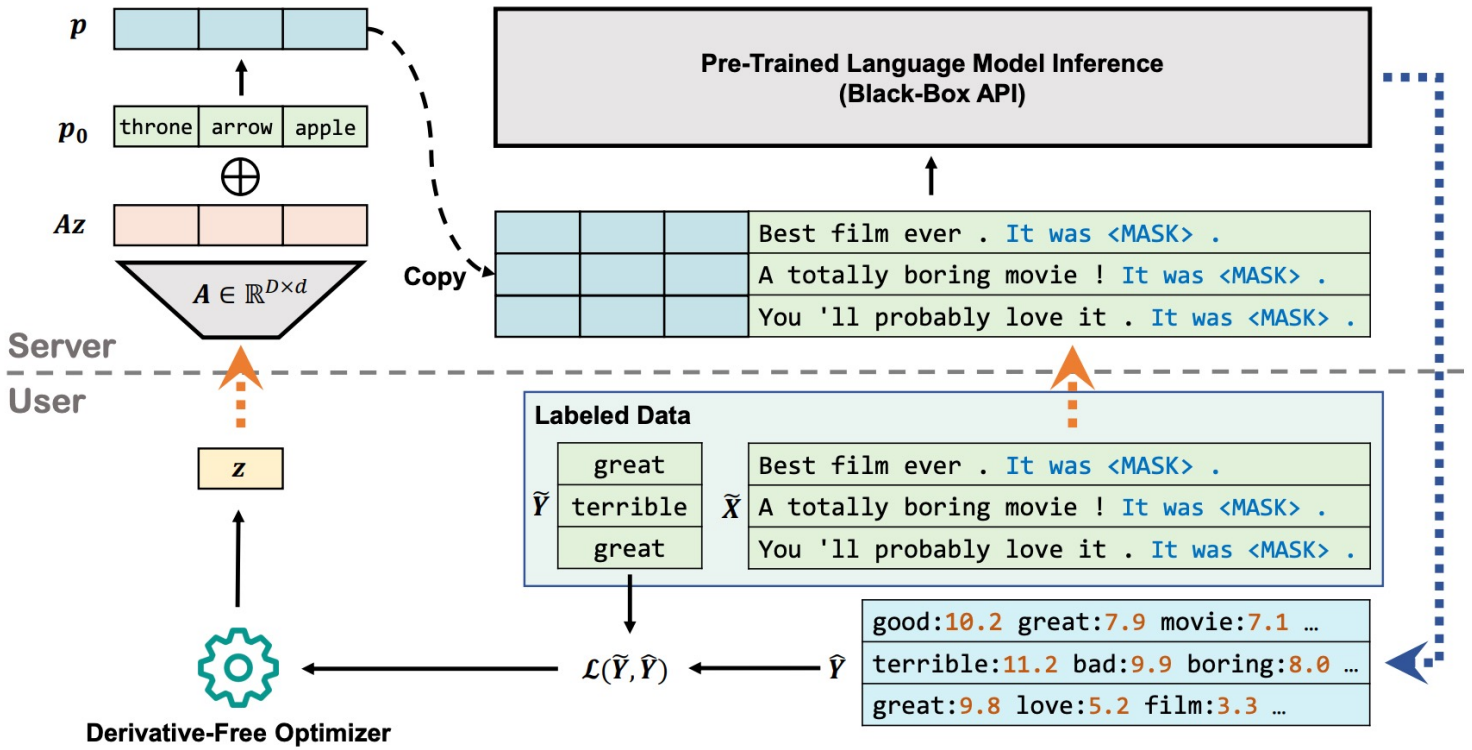
Prompting Methods

- LM-as-a-Service
  - *Black-Box Tuning for Language-Model-as-a-Service* (arXiv 2201.03514)
  - Wrap the input with the template and pass the wrapped input to the large-scale PLM
  - Use the result returned by the large-scale PLM as well as gradient-free methods to optimize the parameters from prompts



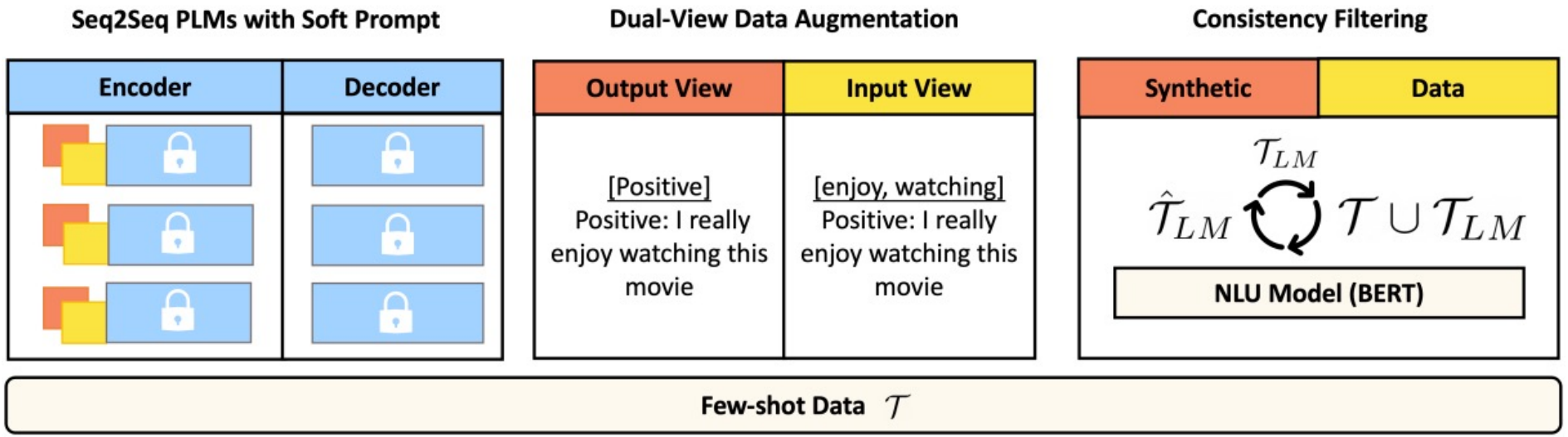
Prompting Methods

- LM-as-a-Service
  - *Black-Box Tuning for Language-Model-as-a-Service* (arXiv 2201.03514)



Recent Applications on Prompt

- Prompt for Data Augmentation
  - *PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks* (ACL 2022)
  - Use prompt as well as the tag to help generating the input example, which can train a better prompt initialization point



Recent Applications on Prompt

- Prompt for Zero-shot Relation Extraction
  - *RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction* (Findings of ACL 2022)
  - Manually design the structured template for relation generator and relation extractor

Input	Relation: <**Label**>.
Example	Relation: **Military Rank**.
Output	Context: <Sentence>. Head Entity: <**Subject**>, Tail Entity: <**Object**>.
Example	Context: Their grandson was Captain Nicolas Tindal. Head Entity: **Nicolas Tindal**, Tail Entity: **Captain**.

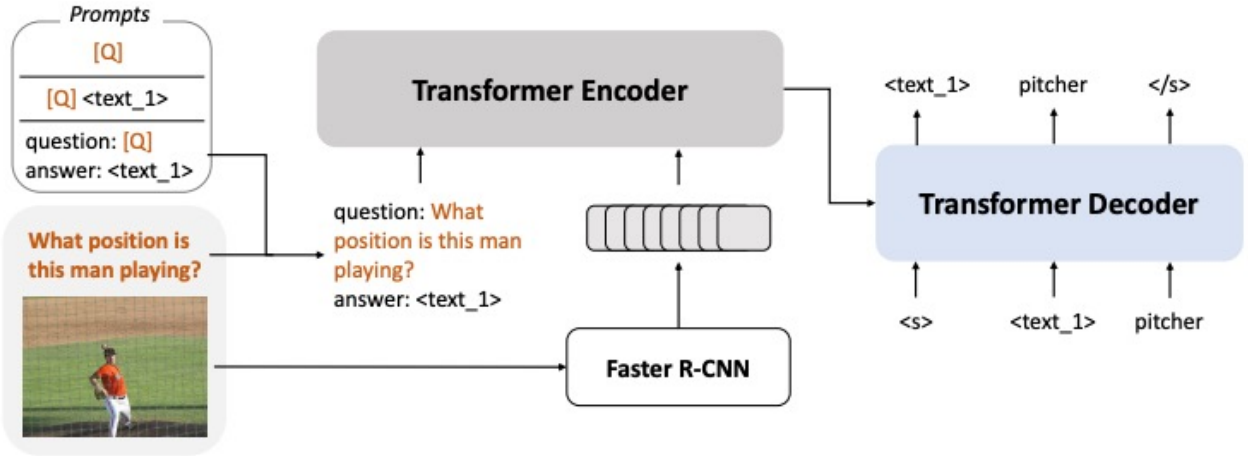
(a) Structured template for relation generator.

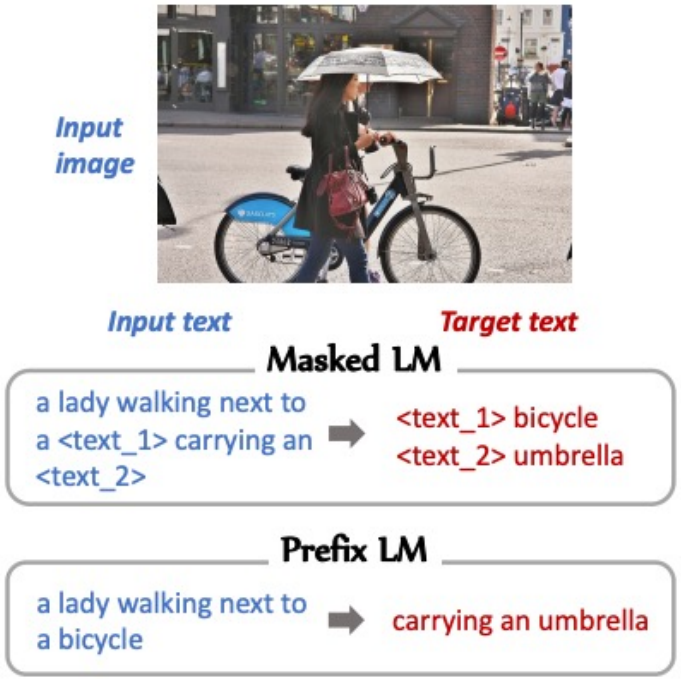
Input	Context: <Sentence>.
Example	Context: Their grandson was Captain Nicolas Tindal.
Output	Head Entity: <**Subject**>, Tail Entity: <**Object**>, Relation: <**Label**>.
Example	Head Entity: **Nicolas Tindal**, Tail Entity: **Captain**, Relation: **Military Rank**.

(b) Structured template for relation extractor.

Recent Applications on Prompt

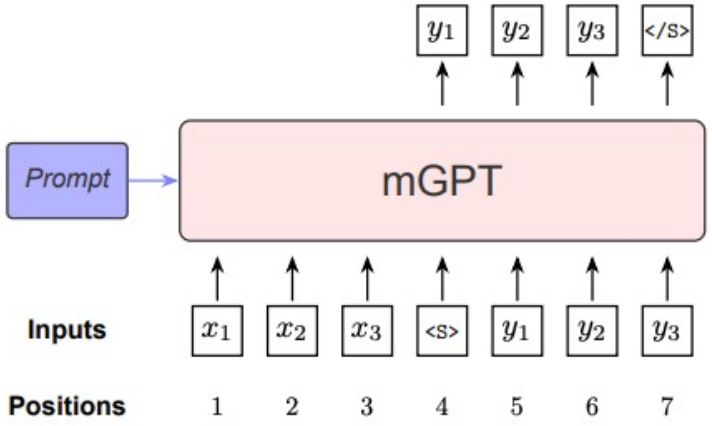
- Vision-Language
  - *A Good Prompt Is Worth Millions of Parameters: Low-resource Prompt-based Learning for Vision-Language Models* (ACL 2022)



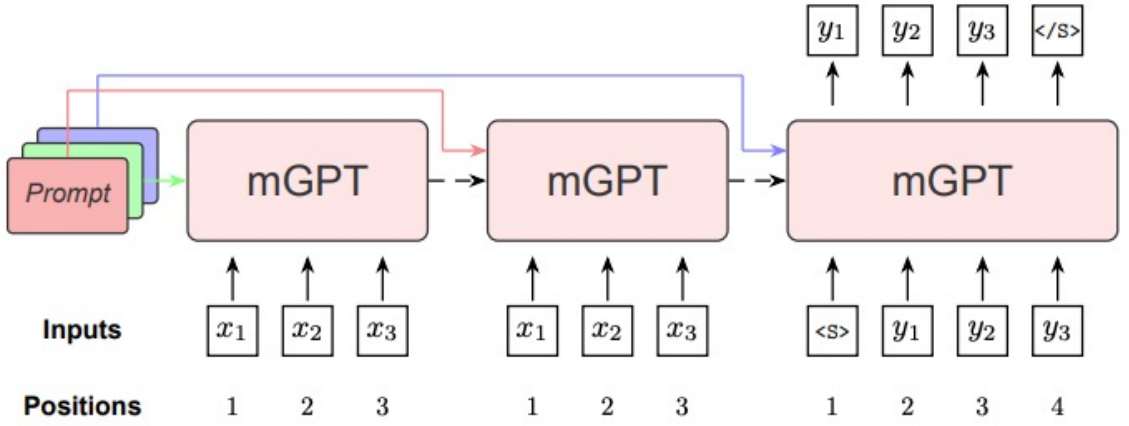


Recent Applications on Prompt

- Machine Translation
  - *MSP: Multi-Stage Prompting for Making Pre-trained Language Models Better Translators* (ACL 2022)



(a) Basic (single-stage) prompting for MT.



(b) Multi-stage prompting.

Thank you