UNIVERSITÄT
ZU KÖLN

In-Context-Learning

HS In Context Learning (ICL) (Summer term 2024)

Nils Reiter,
nils.reiter@uni-koeln.de

May 16, 2024

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

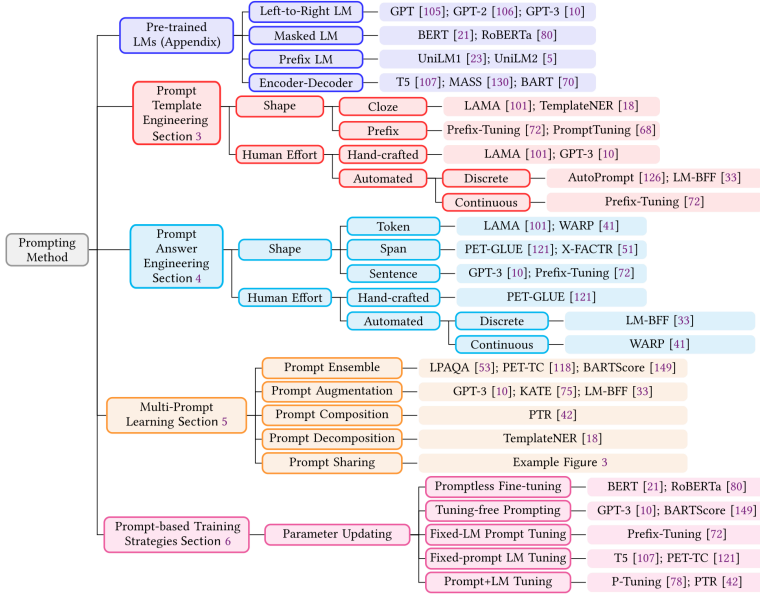
Liu et al. (2023)

Pengfei Liu/Weizhe Yuan/Jinlan Fu/Zhengbao Jiang/Hiroaki Hayashi/Graham Neubig (2023). "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* 55.9. Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3560815. URL: https://doi.org/10.1145/3560815

- First upload on arxiv: July 28, 2021 doi
- First impressions on the text?

Structure

- Two Sea Changes in Natural Language Processing
- A Formal Description of Prompting
- Prompt Template Engineering
- Prompt Answer Engineering
- Multi-prompt Learning
- Training Strategies for Prompting Methods
- Applications
- Prompt-relevant Topics
- Challenges


Prompting Method
Pre-trained LMs (Appendix)
Left-to-Right LM
GPT [105]; GPT-2 [106]; GPT-3 [10]
Masked LM
BERT [21]; RoBERTa [80]
Prefix LM
UniLM1 [23]; UniLM2 [5]
Encoder-Decoder
T5 [107]; MASS [130]; BART [70]
Prompt Template Engineering Section 3
Shape
Cloze
LAMA [101]; TemplateNER [18]
Prefix
Prefix-Tuning [72]; PromptTuning [68]
Human Effort
Hand-crafted
LAMA [101]; GPT-3 [10]
Automated
Discrete
AutoPrompt [126]; LM-BFF [33]
Continuous
Prefix-Tuning [72]
Prompt Answer Engineering Section 4
Shape
Token
LAMA [101]; WARP [41]
Span
PET-GLUE [121]; X-FACTR [51]
Sentence
GPT-3 [10]; Prefix-Tuning [72]
Human Effort
Hand-crafted
PET-GLUE [121]
Automated
Discrete
LM-BFF [33]
Continuous
WARP [41]
Multi-Prompt Learning Section 5
Prompt Ensemble
LPAQA [53]; PET-TC [118]; BARTScore [149]
Prompt Augmentation
GPT-3 [10]; KATE [75]; LM-BFF [33]
Prompt Composition
PTR [42]
Prompt Decomposition
TemplateNER [18]
Prompt Sharing
Example Figure 3
Prompt-based Training Strategies Section 6
Parameter Updating
Promptless Fine-tuning
BERT [21]; RoBERTa [80]
Tuning-free Prompting
GPT-3 [10]; BARTScore [149]
Fixed-LM Prompt Tuning
Prefix-Tuning [72]
Fixed-prompt LM Tuning
T5 [107]; PET-TC [121]
Prompt+LM Tuning
P-Tuning [78]; PTR [42]

A Formal Description of Prompting

- Supervised learning: $P_\theta(y|x)$ (predict output y based on input x and parameters θ)
- Prompting: Use $P_\theta(x)$ to 'derive' y

A Formal Description of Prompting

- Supervised learning: $P_\theta(y|x)$ (predict output y based on input x and parameters θ)
- Prompting: Use $P_\theta(x)$ to 'derive' y
- Three steps
  - Prompt addition: Combine input text x with something to get x' (e.g., apply template)
  - Answer search: Test various possible answers z on x', select the one with highest probability
  - Answer mapping: Map most probable answer z to output y
    - Sometimes trivial

A Formal Description of Prompting

- Supervised learning: $P_\theta(y|x)$ (predict output y based on input x and parameters θ)
- Prompting: Use $P_\theta(x)$ to 'derive' y
- Three steps
  - Prompt addition: Combine input text x with something to get x' (e.g., apply template)
  - Answer search: Test various possible answers z on x', select the one with highest probability
  - Answer mapping: Map most probable answer z to output y
    - Sometimes trivial
- I.e.: What the model really 'knows' is $P_\theta(x)$

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{--, -, \backsim, +, ++\}$

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{--, -, \backsim, +, ++\}$
- Steps
  - Define template as prompting function $f_{\text{prompt}}(\cdot)$: `[X] Overall, it was a [Z] movie.`
  - Prompt addition: Apply f_{prompt}
    $x' =$ `I love this movie. Overall, it was a [Z] movie.`

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{--, -, \backsim, +, ++\}$
- Steps
  - Define template as prompting function $f_{\text{prompt}}(\cdot)$: `[X] Overall, it was a [Z] movie.`
  - Prompt addition: Apply f_{prompt}
    $x' =$ `I love this movie. Overall, it was a [Z] movie.`
  - Answer search: Identify highest ranking $\hat{z}$ to fill into `[Z]`
    - $\mathcal{Z} = \{\text{excellent}, \text{good}, \text{OK}, \text{bad}, \text{horrible}\}$: Permissible values for z
    - $f_{\text{fill}}(x', z)$: Function that fills `[Z]` in x' with z
    - $\hat{z} = \text{search}_{z \in \mathcal{Z}} P_\theta(f_{\text{fill}}(x', z))$
    - E.g.: $\hat{z} =$ `excellent`

Example

Sentiment Analysis (Liu et al., 2023, 3 f.)

- Task definition
  - Input: text $x \in X$, e.g., $x =$ "I love this movie."
  - Output: $\mathcal{Y} = \{--, -, \backsim, +, ++\}$
- Steps
  - Define template as prompting function $f_{\text{prompt}}(\cdot)$: `[X] Overall, it was a [Z] movie.`
  - Prompt addition: Apply f_{prompt}
    $x' =$ `I love this movie. Overall, it was a [Z] movie.`
  - Answer search: Identify highest ranking $\hat{z}$ to fill into `[Z]`
    - $\mathcal{Z} = \{\text{excellent}, \text{good}, \text{OK}, \text{bad}, \text{horrible}\}$: Permissible values for z
    - $f_{\text{fill}}(x', z)$: Function that fills `[Z]` in x' with z
    - $\hat{z} = \text{search}_{z \in \mathcal{Z}} P_\theta(f_{\text{fill}}(x', z))$
    - E.g.: $\hat{z} =$ `excellent`
  - Answer mapping: Map text output to class
    - `excellent` $\rightarrow ++$

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 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.
Answer	$\boldsymbol{z}$	"good," "fantastic," "boring"	A token, phrase, or sentence that fills `[Z]`
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.

$\boldsymbol{z}^*$ represents answers that correspond to true output $\boldsymbol{y}^*$.

Table: Terminology and Notation of Prompting Methods (Liu et al., 2023, 5)

Type	Task Example	Input ([X])	Template	Answer ([Z])
Text Classification	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 Classification	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair Classification	Natural Language Inference	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	Named Entity Recognition	[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. ...
Regression	Textual Similarity	[X1]: A man is smoking. [X2]: A man is skating.	[X1] [Z], [X2]	Yes No ...

Table: Examples of input, template, and answer for Different Tasks (Liu et al., 2023, 5)