

Prompting

By Thanakorn Thaminkaew

Outline


- What is Prompt?
- Why Large Language Model Matters?
- How to Fine-Tune the Language Model?
 - Apply to different tasks
- What are the design considerations for prompt-based methods?
 - Example
- Tools and Additional Resources
- Conclusion


Outline

- **What is Prompt?**
- Why Large Language Model Matters?
- How to Fine-Tune the Language Model?
 - Apply to different tasks
- What are the design considerations for prompt-based methods?
 - Example
- Tools and Additional Resources
- Conclusion

What is Prompt?

ChatGPT

 What is prompt?





A prompt can refer to different things depending on the context. Here are some common meanings:

- 1. In writing: A prompt is a word, phrase, or sentence that is given to a writer to inspire or guide their writing. Prompts are often used in creative writing exercises or as a starting point for a writing assignment.
- 2. In computing: A prompt is a symbol, message, or signal that appears on a computer screen to indicate that the system is waiting for user input. For example, the command prompt in a command-line interface indicates that the system is waiting for a user to enter a command.
- 3. In psychology: A prompt is a cue or stimulus that is given to elicit a specific behavior or response. For example, a therapist might use a prompt to encourage a patient to talk about their feelings or to perform a certain task.
- 4. In education: A prompt is a question or task given to a student to prompt them to think and respond in a particular way. For example, an essay prompt might ask a student to analyze a piece of literature or to argue for a particular point of view.

What is Prompt?

ChatGPT

 What is prompt?



A prompt can refer to different things depending on the context. Here are some common meanings:

- 1. In writing: A prompt is a word, phrase, or sentence that is given to a writer to inspire or guide their writing. Prompts are often used in creative writing exercises or as a starting point for a writing assignment.
- 2. In computing: A prompt is a symbol, message, or signal that appears on a computer screen to indicate that the system is waiting for user input. For example, the command prompt in a command-line interface indicates that the system is waiting for a user to enter a command.
- 3. In psychology: A prompt is a cue or stimulus that is given to elicit a specific behavior or response. For example, a therapist might use a prompt to encourage a patient to talk about their feelings or to perform a certain task.
- 4. In education: A prompt is a question or task given to a student to prompt them to think and respond in a particular way. For example, an essay prompt might ask a student to analyze a piece of literature or to argue for a particular point of view.

What is Prompt?

ChatGPT



What is prompt in context of NLP?



In the context of natural language processing (NLP), a prompt is a piece of text used to initiate a task or query to a language model. It is a specific text input that is given to the language model to generate a response, and it can be a sentence, a paragraph, or a set of keywords that provide context and direction for the model to produce relevant output.



For instance, in the task of text generation, a prompt could be the beginning of a sentence, and the model will continue the sentence based on its understanding of language patterns and associations. In question answering, a prompt could be a question, and the model will generate an answer based on its knowledge of the topic.

Prompts can be manually crafted by humans or generated automatically using various techniques such as keyword extraction or clustering. They are a crucial component of NLP tasks and are used to guide the language model to produce specific output that aligns with the task at hand.

What is Prompt?

ChatGPT



What is prompt in context of NLP?



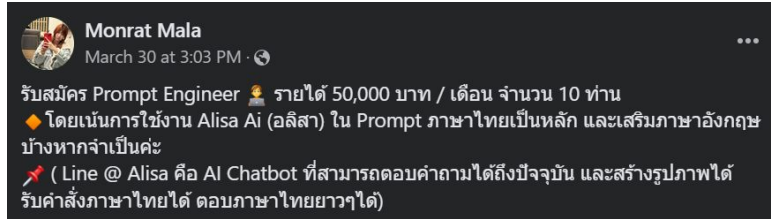
In the context of natural language processing (NLP), a prompt is a piece of text used to initiate a task or query to a language model. It is a specific text input that is given to the language model to generate a response, and it can be a sentence, a paragraph, or a set of keywords that provide context and direction for the model to produce relevant output.



For instance, in the task of text generation, a prompt could be the beginning of a sentence, and the model will continue the sentence based on its understanding of language patterns and associations. In question answering, a prompt could be a question, and the model will generate an answer based on its knowledge of the topic.

Prompts can be manually crafted by humans or generated automatically using various techniques such as keyword extraction or clustering. They are a crucial component of NLP tasks and are used to guide the language model to produce specific output that aligns with the task at hand.

What is Prompt?



<https://web.facebook.com/groups/5779011592182423/permalink/6080231392060440/>

คุณสมบัติ

- เคยมีประสบการณ์ใช้งาน Generative Ai ทั้งในฝั่งของ Text และ Image และสามารถออกแบบคำสั่งและปรับจูนคำสั่งให้ได้ Output ในแบบที่ต้องการได้อย่างเชี่ยวชาญ และสามารถแนะนำผู้อื่นให้เข้าใจได้
 - หากเคยใช้ Generative Ai ในการหารายได้ จะพิจารณาเป็นพิเศษ
 - เข้าใจการใช้งาน Multiple Routes Session Memory และการใช้งานแบบหลาย Concurrent ของ Alisa Ai
 - มีไอเดียในการออกแบบและใช้งาน Generative Ai ในการสร้างรายได้ , ลดค่าใช้จ่าย , ประหยัดเวลาในการทำงาน ช่วยในการทำงานอื่นๆ ออกแบบการใช้ Generative Ai เพื่อปรับปรุงการทำงานให้มีประสิทธิภาพมากขึ้นได้
 - มีความสามารถในการเขียน และทำ Graphic จานจงงานด้วยตนเองได้ (*สำคัญ)
 - หากสามารถเขียนโปรแกรมได้ และสามารถนำความรู้มาใช้กับงาน Dev พร้อมสอนคนอื่นๆได้ จะพิจารณาเป็นพิเศษ(และพิจารณารายได้เพิ่มเติม)
 - ไม่จำกัดวุฒิการศึกษา
- รายละเอียดงาน
- เน้นทำ Research , Report , บทความ , ผลงาน , ชิ้นงาน ในแง่ของการนำ Generative Ai ไปสร้างรายได้ ลดค่าใช้จ่าย หรือ ประหยัดเวลา/เพิ่มประสิทธิภาพการทำงาน
 - *โดยทาง Prompt Engineer จะต้องสามารถออกแบบและแนะนำได้ด้วยตนเองเป็นหลักว่าควรทำอะไร/ทำอะไรได้บ้าง (แต่อาจจะมียางกรณีที่เป็น Request จากฝั่งบริษัทเป็นเคสๆ ไปค่ะ)

What is Prompt?

What is Prompt Engineering and Why It Might Be Your Next Career Path



“There are really **two categories** I define in prompt engineering, **[1] defining queries to existing AI models**, like OpenAI or MidJourney, and the engineering required to **[2] build and train the models**. Both of these classifications will see a sharp increase in demand for professionals to help businesses leverage Artificial intelligence for internal tools or products and services.”

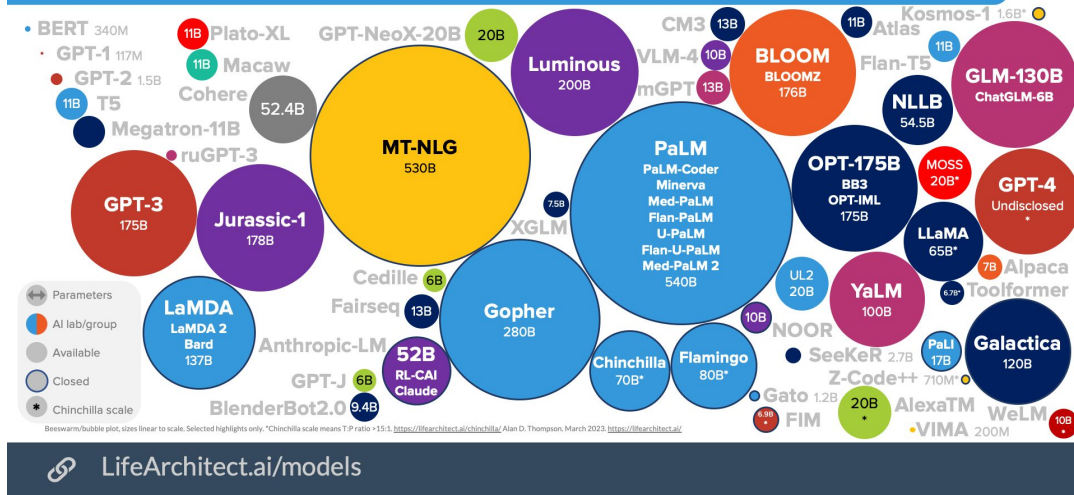
Outline

- What is Prompt?
- **Why Large Language Model Matters?**
- How to Fine-Tune the Language Model?
 - Apply to different tasks
- What are the design considerations for prompt-based methods?
 - Example
- Tools and Additional Resources
- Conclusion

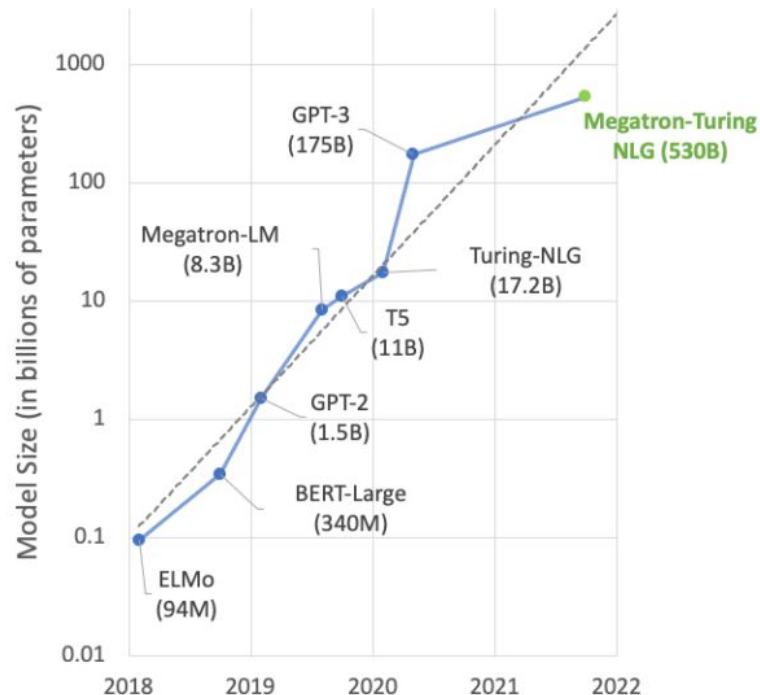
Why Large Language Model Matters?

Current Language Model Sizes

LANGUAGE MODEL SIZES TO MAR/2023



Currently, no specifications are displayed regarding the parameters used in GPT-4. Although, there were speculations that OpenAI has used around 100 Trillion parameters for GPT-4. This was later denied by OpenAI CEO Sam Altman. But since GPT-3 has 175 billion parameters added we can expect a higher number on this new language model GPT-4.



Why Large Language Model Matters?

Language Models are Few-Shot Learners

Tradition Fine-Tuning (BERT)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



Why Large Language Model Matters?

Language Models are Few-Shot Learners

Tradition Fine-Tuning (BERT)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



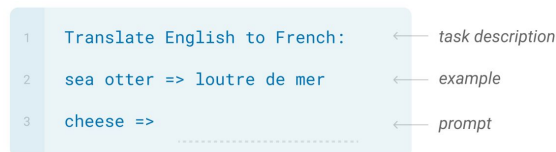
In-Context Learning (GPT-3)

Zero-Shot

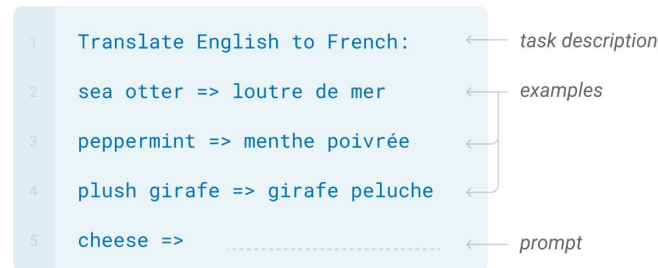
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



One-Shot



Few-Shot



Why Large Language Model Matters?

Language Models are Few-Shot Learners

Tradition Fine-Tuning (BERT)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



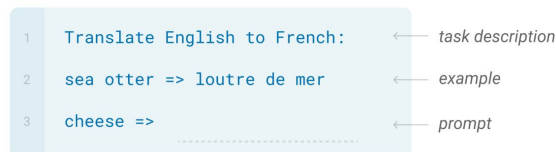
In-Context Learning (GPT-3)

Zero-Shot

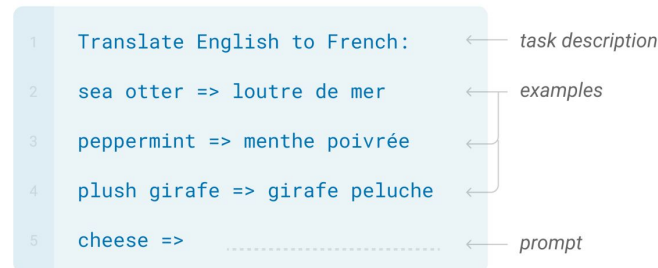
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



One-Shot



Few-Shot



In-Context Learning (GPT-3)

Model

- No Fine-Tune

Prompt

- Task Description
- Example
- Prompt

Why Large Language Model Matters?

Language Models are Few-Shot Learners

Dataset: BoolQ

Context → Normal force -- In a simple case such as an object resting upon a table, the normal force on the object is equal but in opposite direction to the gravitational force applied on the object (or the weight of the object), that is, $N = m g$ ($\displaystyle N=mg$), where m is mass, and g is the gravitational field strength (about 9.81 m/s on Earth). The normal force here represents the force applied by the table against the object that prevents it from sinking through the table and requires that the table is sturdy enough to deliver this normal force without breaking. However, it is easy to assume that the normal force and weight are action-reaction force pairs (a common mistake). In this case, the normal force and weight need to be equal in magnitude to explain why there is no upward acceleration of the object. For example, a ball that bounces upwards accelerates upwards because the normal force acting on the ball is larger in magnitude than the weight of the ball.
question: is the normal force equal to the force of gravity?
answer:

Target Completion → yes

Why Large Language Model Matters?

Language Models are Few-Shot Learners

Zero-Shot

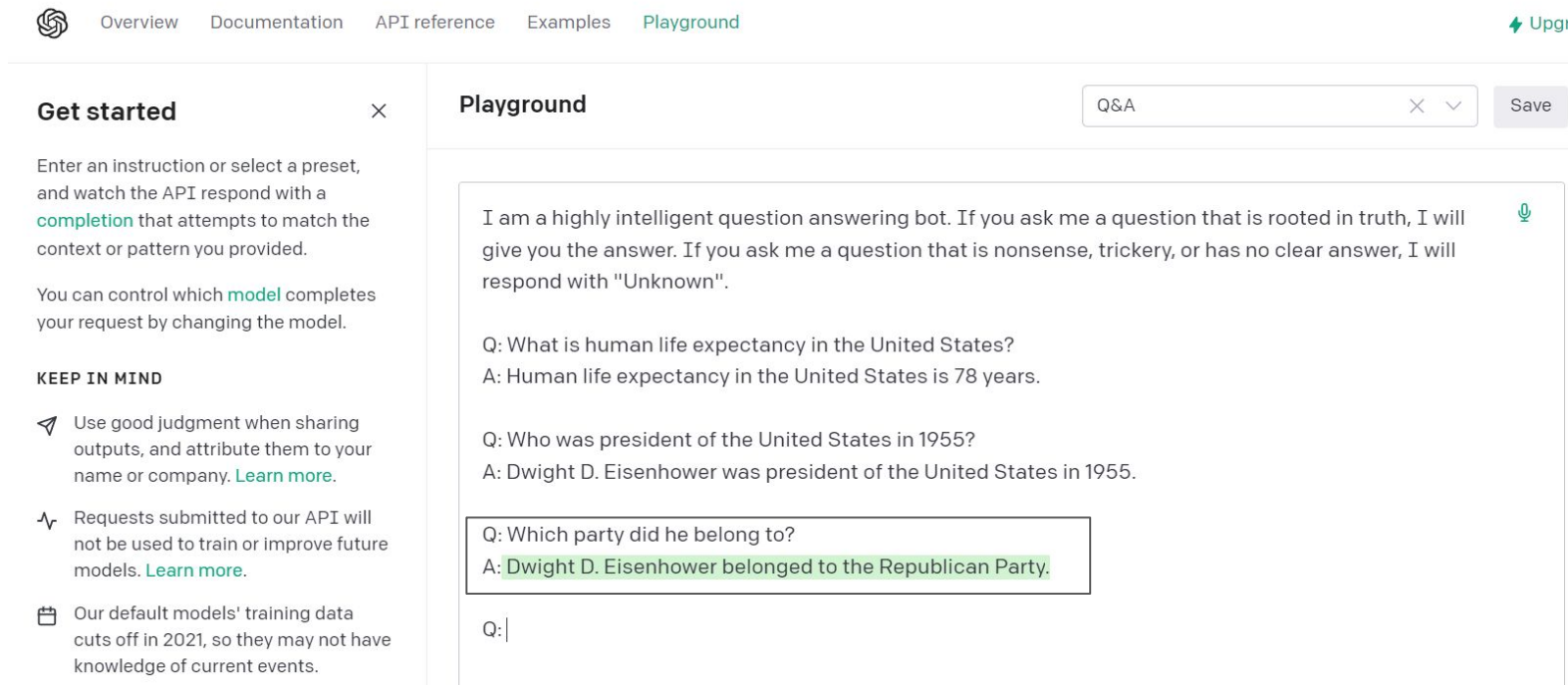
Dataset: BoolQ

Context →	<p>Normal force -- In a simple case such as an object resting upon a table, the normal force on the object is equal but in opposite direction to the gravitational force applied on the object (or the weight of the object), that is, $N = m g$ ($\displaystyle N=mg$), where m is mass, and g is the gravitational field strength (about 9.81 m/s on Earth). The normal force here represents the force applied by the table against the object that prevents it from sinking through the table and requires that the table is sturdy enough to deliver this normal force without breaking. However, it is easy to assume that the normal force and weight are action-reaction force pairs (a common mistake). In this case, the normal force and weight need to be equal in magnitude to explain why there is no upward acceleration of the object. For example, a ball that bounces upwards accelerates upwards because the normal force acting on the ball is larger in magnitude than the weight of the ball.</p>	Text Input
Prompt	<p>question: is the normal force equal to the force of gravity? answer:</p>	
Target Completion →	yes	

Why Large Language Model Matters?

GPT Demo: <https://platform.openai.com/playground>

Few-Shot



The screenshot shows the OpenAI Playground interface. On the left is a sidebar with navigation links: Overview, Documentation, API reference, Examples, and Playground (highlighted). Below these is a 'Get started' section with instructions on how to use the playground and a 'KEEP IN MIND' section with three important notes. The main area is titled 'Playground' and contains a Q&A input field, a 'Save' button, and a text area showing a few-shot example. The example consists of a system prompt, two Q&A pairs, and a third Q&A pair where the answer is highlighted in green.

Get started ×

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more.](#)
- Requests submitted to our API will not be used to train or improve future models. [Learn more.](#)
- Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

Playground Q&A × ▾ Save

I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".

Q: What is human life expectancy in the United States?
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?
A: Dwight D. Eisenhower was president of the United States in 1955.



Q: Which party did he belong to?
A: Dwight D. Eisenhower belonged to the Republican Party.

Q: |

Why Large Language Model Matters?

GPT Demo: <https://platform.openai.com/playground>

Few-Shot

 Overview Documentation API reference Examples Playground  Upgrade

Get started

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more.](#)
- Requests submitted to our API will not be used to train or improve future models. [Learn more.](#)
- Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

Playground

Q&A × ▼ Save

A great movie!!!

Q: Is it positive or negative?

A: positive

###

The storyline is boring.

Q: Is it positive or negative?

A: negative

###

The movie is emotional.

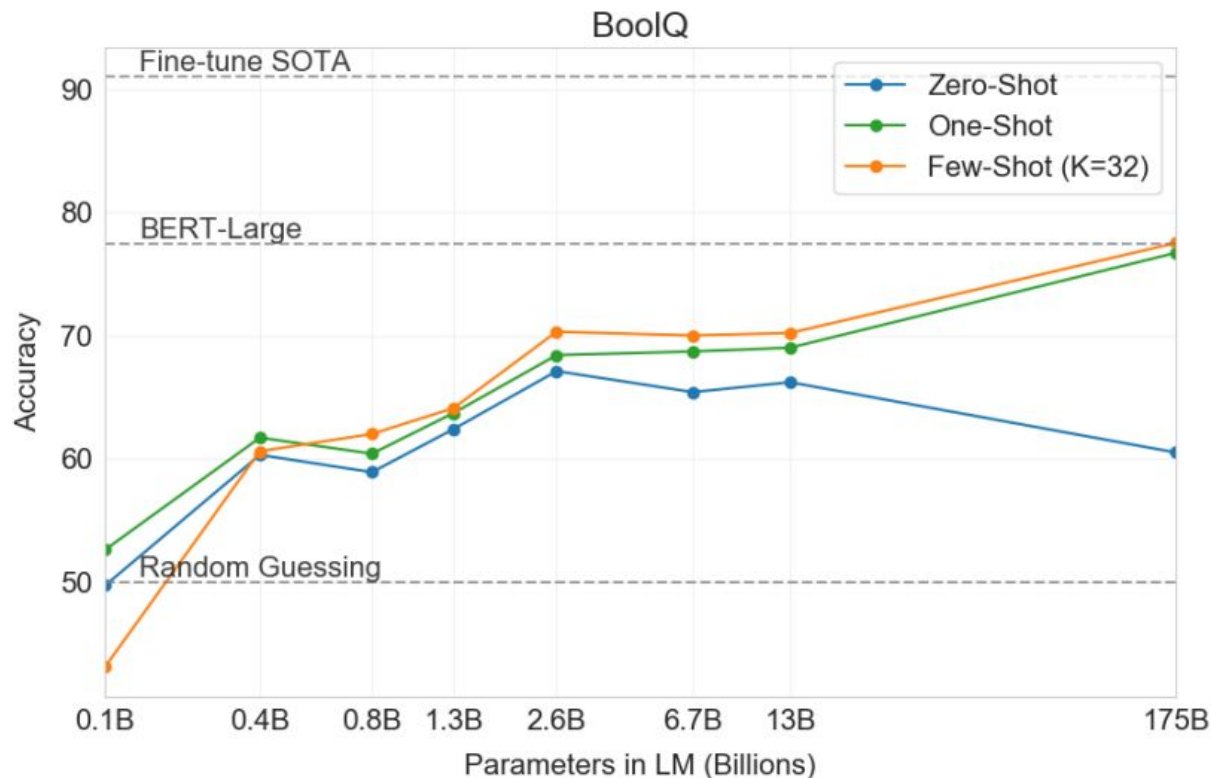
Q: Is it positive or negative?

A: **positive**

Q: |

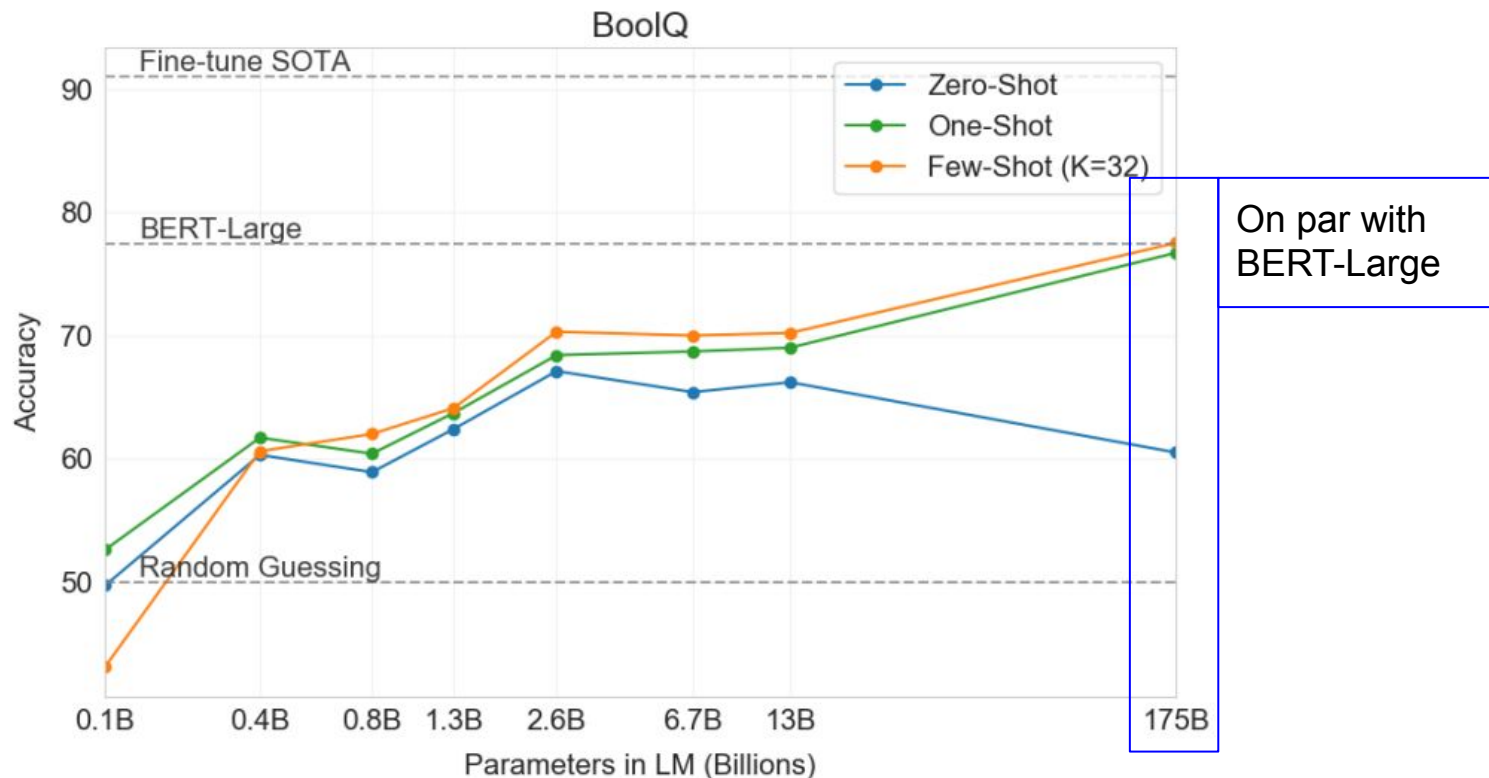
Why Large Language Model Matters?

Language Models are Few-Shot Learners



Why Large Language Model Matters?

Language Models are Few-Shot Learners



Why Large Language Model Matters?

Language Models are Few-Shot Learners

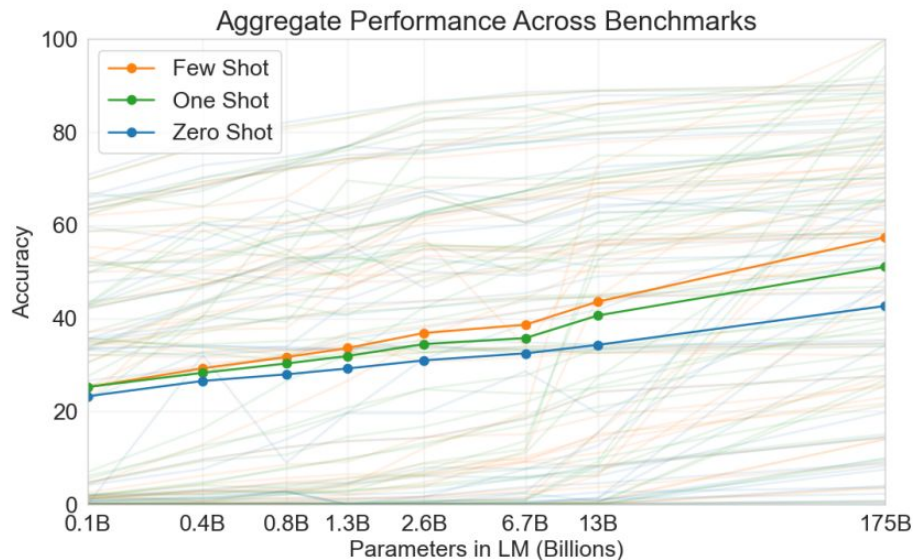
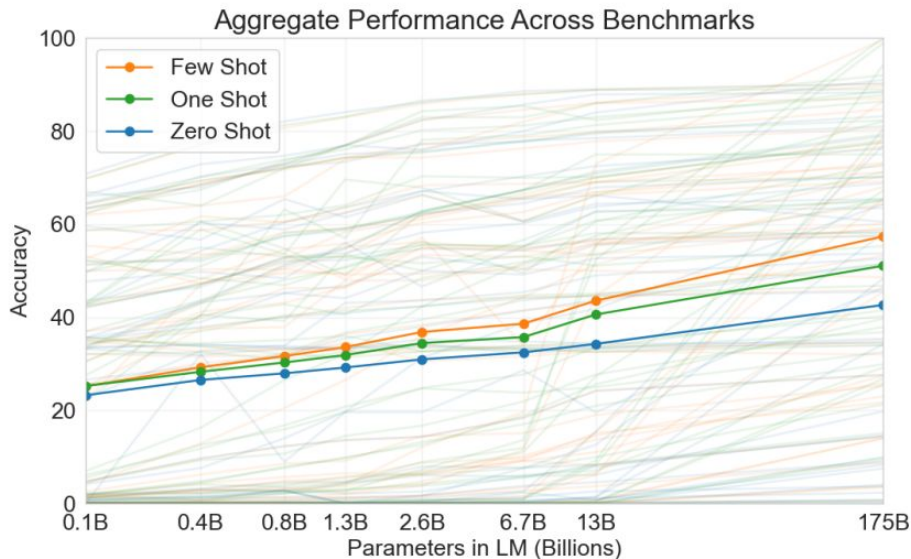


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

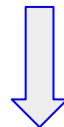
Why Large Language Model Matters?

Language Models are Few-Shot Learners



General Model:

On average, the accuracy in most tasks are not as good



Fine-Tune

Specific Model:

Fine-tune model towards a specific task

Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

Outline

- What is Prompt?
- Why Large Language Model Matters?
- **How to Fine-tune the Language Model?**
 - Apply to different tasks
- What are the design considerations for prompt-based methods?
 - Example
- Tools and Additional Resources
- Conclusion

How to Fine-tune the Language Model?

The Power of Scale for Parameter-Efficient Prompt Tuning

Prompt Design

Input Text: A great movie!!!

Prompt: Q: Is it positive or negative?
A:

Freeze



Pre-trained
Model

Freeze

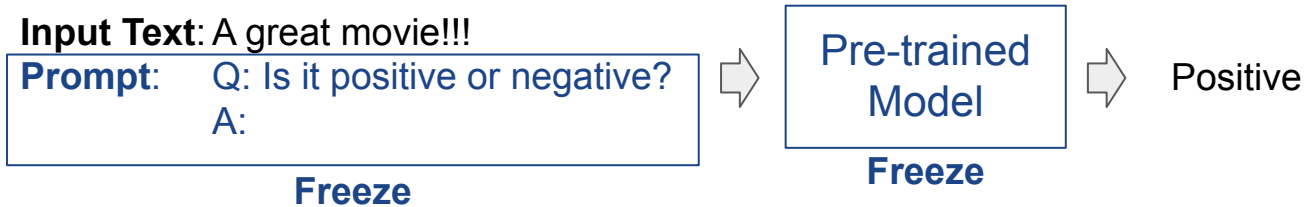


Positive

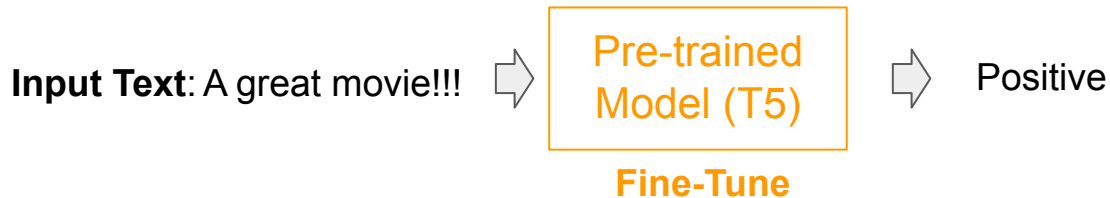
How to Fine-tune the Language Model?

The Power of Scale for Parameter-Efficient Prompt Tuning

Prompt Design



Model Tuning/Full Fine-Tuning



How to Fine-tune the Language Model?

The Power of Scale for Parameter-Efficient Prompt Tuning

Prompt Design

Input Text: A great movie!!!

Prompt: Q: Is it positive or negative?
A:

Freeze



Pre-trained
Model



Positive

Freeze

Model Tuning/Full Fine-Tuning

Input Text: A great movie!!!



Pre-trained
Model (T5)



Positive

Fine-Tune

Prompt Tuning

Fine-Tune

<random
vector>

Input Text:
A great movie !!!



Pre-trained
Model

Freeze

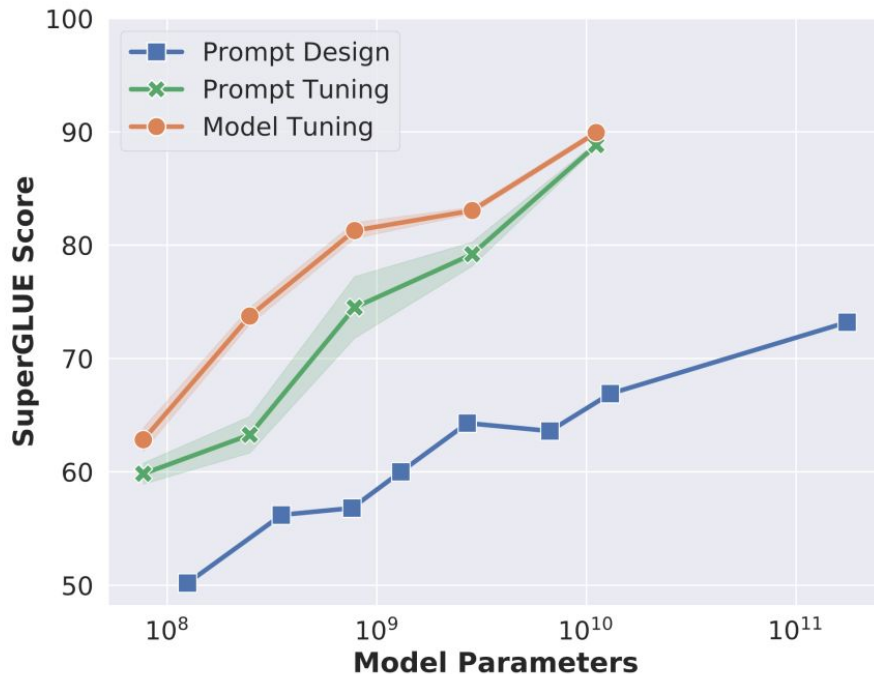
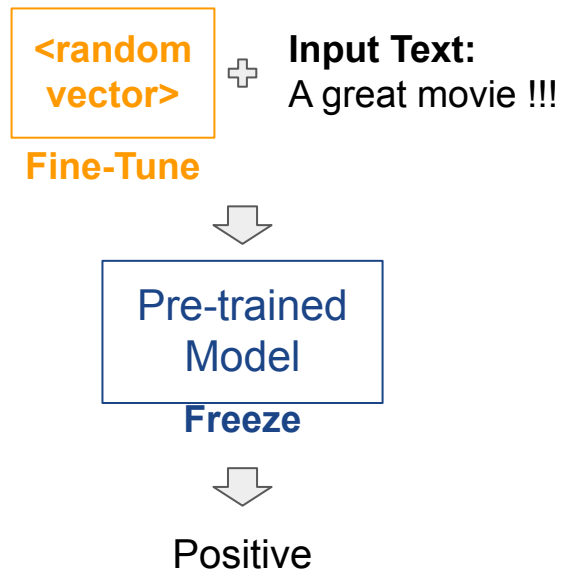


Positive

How to Fine-tune the Language Model?

The Power of Scale for Parameter-Efficient Prompt Tuning

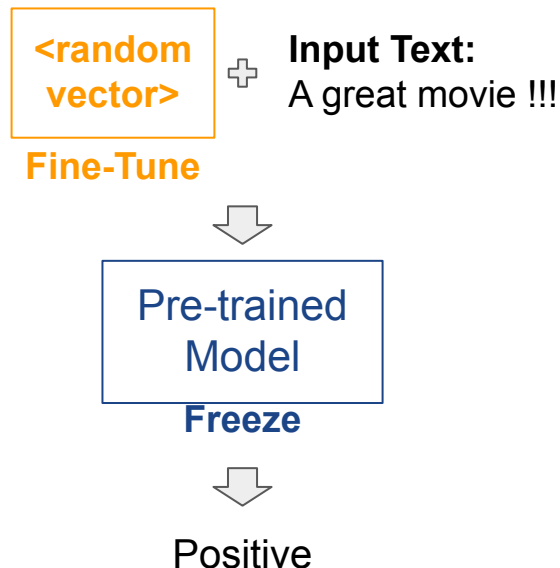
Prompt Tuning



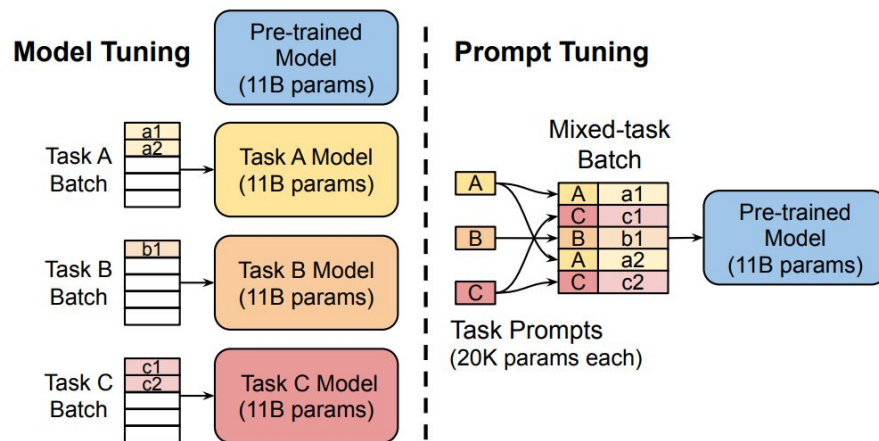
How to Fine-tune the Language Model?

The Power of Scale for Parameter-Efficient Prompt Tuning

Prompt Tuning



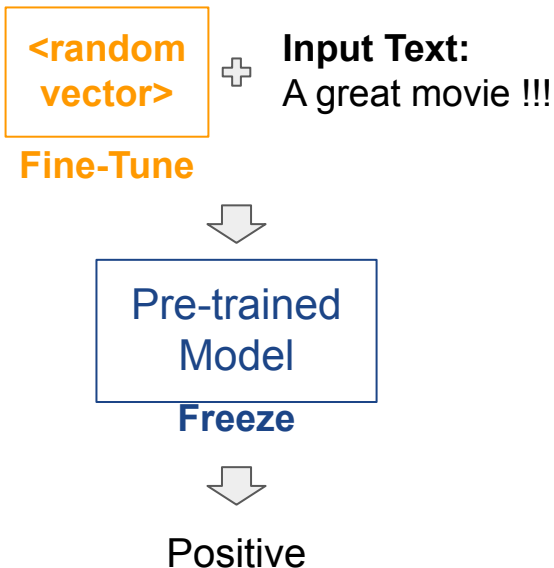
- Fine-tuning with less parameters
- Storing a small task-specific prompt for each task



How to Fine-tune the Language Model?

The Power of Scale for Parameter-Efficient Prompt Tuning

Prompt Tuning



In-Context Learning (GPT-3) / Prompt Design

Model

- No Fine-Tune

Prompt

- Task Description
- Example
- Prompt



Prompt Tuning

Model

- No Fine-Tune

Prompt

- Add <random vector> before input text

Outline

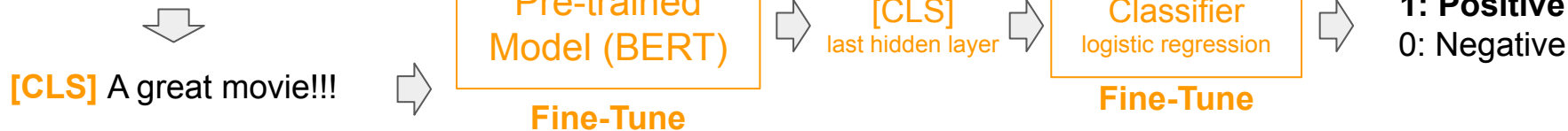
- What is Prompt?
- Why Large Language Model Matters?
- How to Fine-tune the Language Model?
 - **Apply to different tasks**
- What are the design considerations for prompt-based methods?
 - Example
- Tools and Additional Resources
- Conclusion

How to Fine-tune the Language Model?

Sentiment Analysis (Binary Classification)

Model Tuning/Full Fine-Tuning

Input Text: A great movie!!!



How to Fine-tune the Language Model?

Sentiment Analysis (Binary Classification)

Prompt Tuning

Input Text: A great movie!!!



Template/Prompt Engineering

<random vector> + [Input Text] It was <mask>

Verbalizer/Answer Engineering

Positive: Great Negative: Terrible

How to Fine-tune the Language Model?

Sentiment Analysis (Binary Classification)

Prompt Tuning

Input Text: A great movie!!!



Template/Prompt Engineering

<random vector> + [Input Text] It was <mask>



<random vector> + A great movie!!! It was <mask>



Verbalizer/Answer Engineering

Positive: Great Negative: Terrible

Pre-trained
Model (BERT)

Freeze

How to Fine-tune the Language Model?

Sentiment Analysis (Binary Classification)

Prompt Tuning

Input Text: A great movie!!!



Template/Prompt Engineering

<random vector> + [Input Text] It was <mask>



<random vector> + A great movie!!! It was <mask>



Verbalizer/Answer Engineering

Positive: Great Negative: Terrible



Pre-trained
Model (BERT)

Freeze



Great (0.7)
Terrible (0.3)

How to Fine-tune the Language Model?

Sentiment Analysis (Binary Classification)

Model Tuning/Full Fine-Tuning

Input Text: A great movie!!!



Prompt Tuning

Input Text: A great movie!!!



Why Template and Verbalizer Matters?

Making Pre-trained Language Models Better Few-shot Learners

Different Template

Template	Label words	Accuracy
SNLI (entailment/neutral/contradiction)		mean (std)
$\langle S_1 \rangle ? [\text{MASK}] , \langle S_2 \rangle$	Yes/Maybe/No	77.2 (3.7)
$\langle S_1 \rangle . [\text{MASK}] , \langle S_2 \rangle$	Yes/Maybe/No	76.2 (3.3)
$\langle S_1 \rangle ? [\text{MASK}] \langle S_2 \rangle$	Yes/Maybe/No	74.9 (3.0)
$\langle S_1 \rangle \langle S_2 \rangle [\text{MASK}]$	Yes/Maybe/No	65.8 (2.4)
$\langle S_2 \rangle ? [\text{MASK}] , \langle S_1 \rangle$	Yes/Maybe/No	62.9 (4.1)
$\langle S_1 \rangle ? [\text{MASK}] , \langle S_2 \rangle$	Maybe/No/Yes	60.6 (4.8)
Fine-tuning	-	48.4 (4.8)

Different Verbalizer

Template	Label words	Accuracy
SST-2 (positive/negative)		mean (std)
$\langle S_1 \rangle$ It was $[\text{MASK}]$.	great/terrible	92.7 (0.9)
$\langle S_1 \rangle$ It was $[\text{MASK}]$.	good/bad	92.5 (1.0)
$\langle S_1 \rangle$ It was $[\text{MASK}]$.	cat/dog	91.5 (1.4)
$\langle S_1 \rangle$ It was $[\text{MASK}]$.	dog/cat	86.2 (5.4)
$\langle S_1 \rangle$ It was $[\text{MASK}]$.	terrible/great	83.2 (6.9)
Fine-tuning	-	81.4 (3.8)

Other Tasks

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

Type	Task	Text Input [X]	Template	Verbalizer [Z]
Text Classification	Sentiment	A great movie!!!	[X] It was [Z]	Great, Terrible, ...
Text-pair Classification	Natural Language Inference (NLI)	[X1] An old man with ... [X2] A man walks ...	[X1] ? [Z], [X2]	Yes, No, ...
Tagging	Named Entity Recognition (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	ฉันรักคุณ	Thai: [X] English: [Z]	I love you., I fancy you., ...

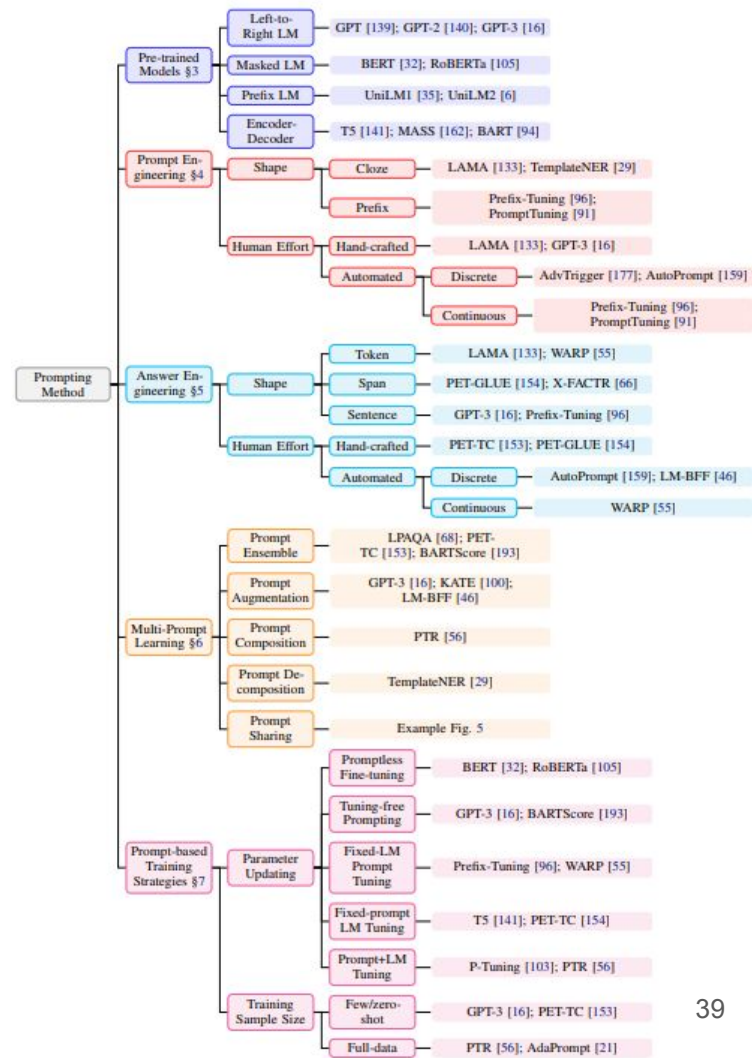
Outline

- What is Prompt?
- Why Large Language Model Matters?
- How to Fine-tune the Language Model?
 - Apply to different tasks
- **What are the design considerations for prompt-based methods?**
 - Example
- Tools and Additional Resources
- Conclusion

Prompt Design Considerations

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

1. Pre-trained Model
2. Prompt Engineering (Template)
3. Answer Engineering (Verbalizer)
4. Expanding the Paradigm (Multi-Prompt Learning)
5. Training Strategies

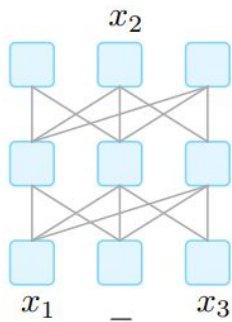


Prompt Design Considerations:

1. Pre-trained Model

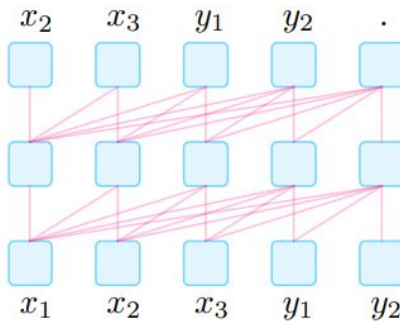
Given a task, which pre-trained language model would be the most appropriate one?

Masked LM
(Encoder)



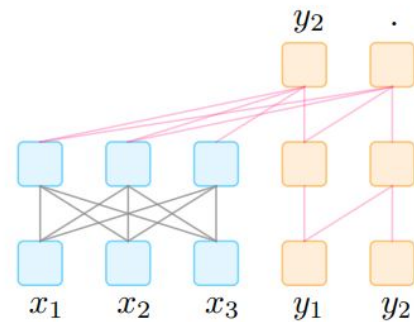
- BERT
- RoBERTa
- WangchanBERTa

Left-to-Right LM
(Decoder)



- GPT
- GPT-2
- GPT-3

Encoder-Decoder



- T5
- mT5
- BART

Prompt Design Considerations:

2. Prompt Engineering (Template)

How to define appropriate prompt templates?

Input Text: A great movie!!!

Template: This review is <mask>

- Human Effect (Prompt word)
 - Hand-craft/ Manual
 - A great movie!!! **This review is <mask>.**
 - Automated
 - **Discrete:** A great movie!!! **<search word> <mask>.**
=> A great movie!!! **It is <mask>.**
 - **Continuous:** A great movie!!! **<search vector> <mask>**
=> A great movie!!! **<random vector> <mask>**

Prompt Design Considerations:

3. Answer Engineering (Verbalizer)

Given a task, how to define a mapping function between label and answer space?

Input Text: A great movie!!!

Answer: This review is <mask>

- Shape
 - Token
 - Positive, Bad, Politics
 - Span/ Sentences
 - I was born in 1995.
- Human Effect (Prompt word)
 - Hand-craft/ Manual
 - Positive: Great Negative: Terrible
 - Automated
 - Discrete: Positive: <search word> Negative: <search word>
=> Positive: Good Negative: Boring
 - Continuous: Positive: <search vector> Negative: <search vector>
=> Positive: <random vector> Negative: <random vector>

Prompt Design Considerations:

4. Expanding the Paradigm (Multi-Prompt Learning)

How to extend the current prompting framework to support more NLP tasks?

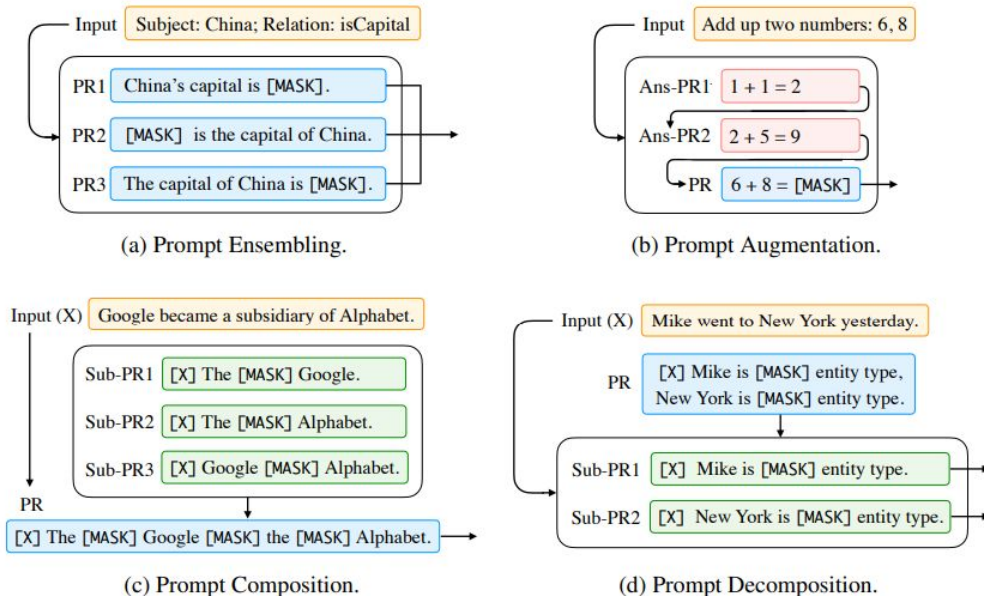
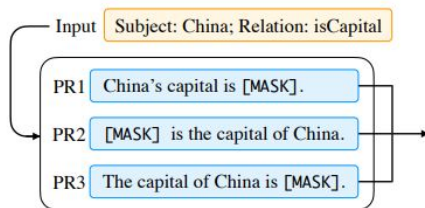


Figure 4: Different multi-prompt learning strategies. We use different colors to differentiate different components as follows. “ ” for input text, “ ” for prompt, “ ” for answered prompt. “ ” for sub-prompt. We use the following abbreviations. “PR” for prompt, “Ans-PR” for answered prompt, “Sub-PR” for sub-prompt.

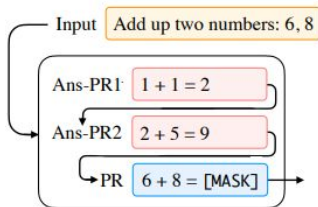
Prompt Design Considerations:

4. Expanding the Paradigm (Multi-Prompt Learning)

How to extend the current prompting framework to support more NLP tasks?



(a) Prompt Ensembling.



(b) Prompt Augmentation.

Prompt Ensembling

- Using multiple unanswered prompts for an input at inference time to make predictions
- Advantages
 - Utilize complementary advantages
 - Stabilize performance on downstream tasks

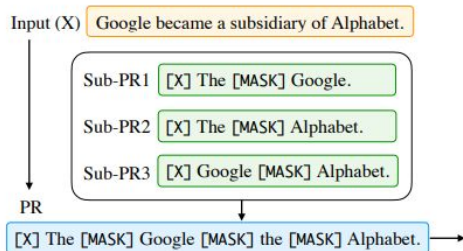
Prompt Augmentation

- Help the model answer the prompt with additional answered prompts
- Advantages
 - make use of the small amount of information that has been annotated
- Core step
 - Selecting & Ordering of answered prompts

Prompt Design Considerations:

4. Expanding the Paradigm (Multi-Prompt Learning)

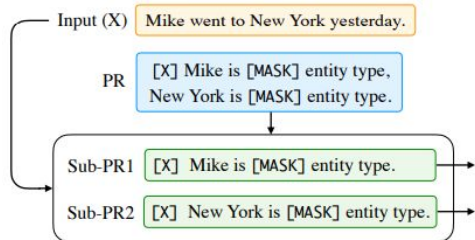
How to extend the current prompting framework to support more NLP tasks?



(c) Prompt Composition.

Prompt Composition

- Prompts for a composable task can be designed with multiple sub-prompts, which can then be combined to complete the task
- Advantages
 - It provides a method of prompt learning for complex tasks



(d) Prompt Decomposition.

Prompt Decomposition

- For tasks where multiple predictions should be performed for one sample, handle it individually
- Advantages
 - Break-down a complicated task into multiple separate ones

Prompt Design Considerations:

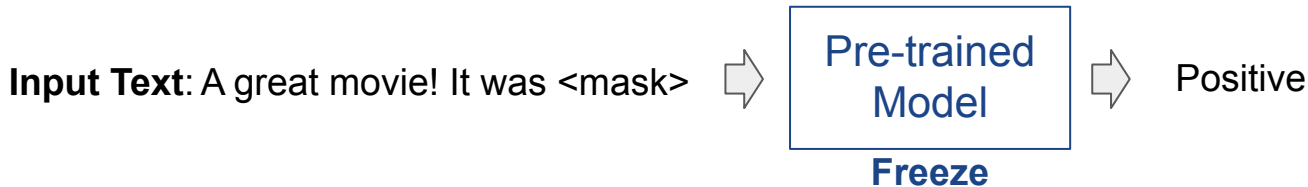
5. Training Strategies

5.1 Promptless Fine-tuning

Input Text: A great movie!



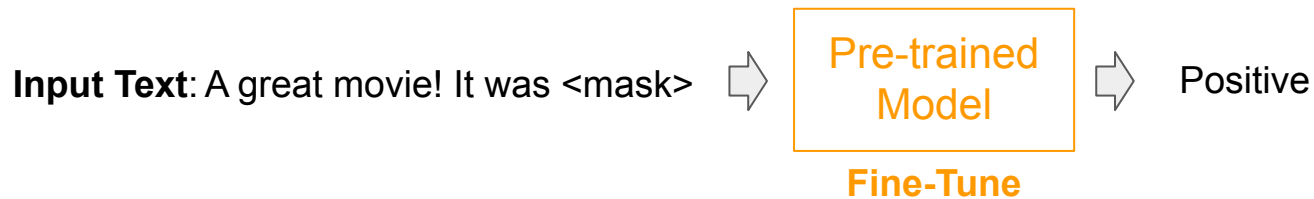
5.2 Tuning-free Prompting



Prompt Design Considerations:

5. Training Strategies

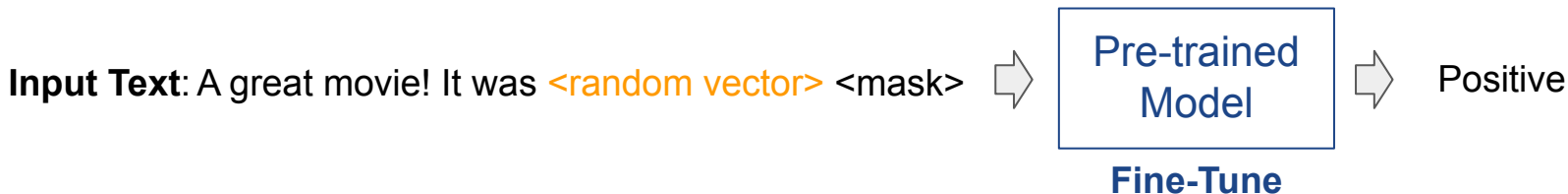
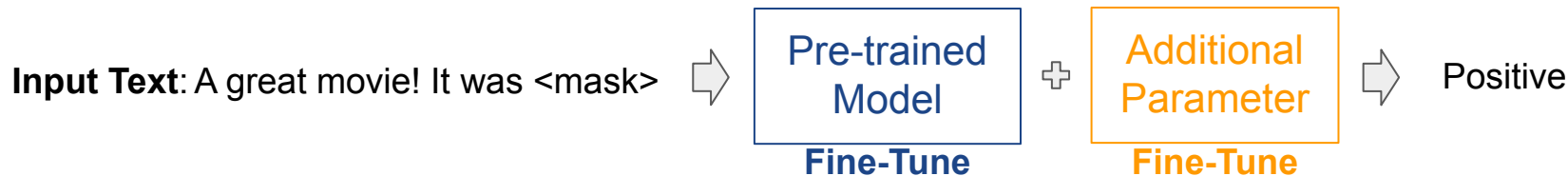
5.3 Fixed-prompt LM Tuning



Prompt Design Considerations:

5. Training Strategies

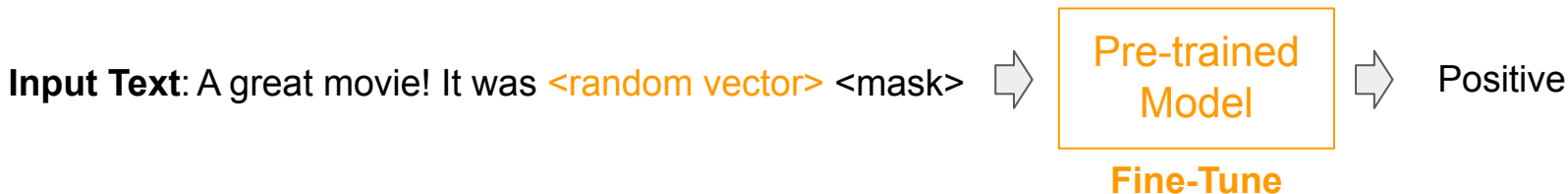
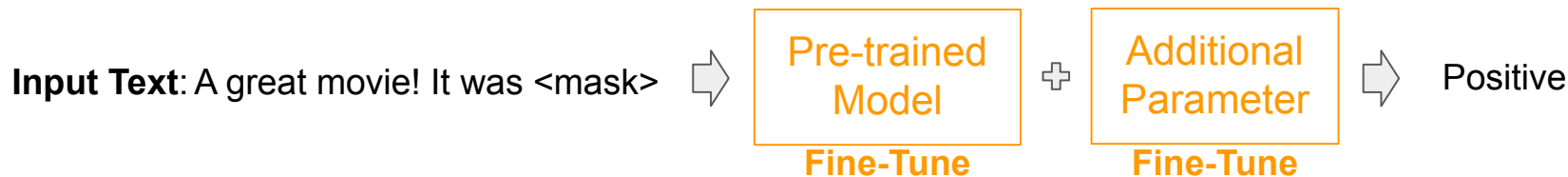
5.4 Fixed-LM Prompt Tuning



Prompt Design Considerations:

5. Training Strategies








5.5 Prompt+LM Fine-tuning



Prompt Design Considerations:

5. Training Strategies

= Freeze LM parameters
 = Finetune LM parameter
 = <random vector>

Strategy	LM Params	Prompt Parameters		Current Model	Example
		Additional	Tuned		
Promptless Fine-tuning	Tuned	-	-	BERT, T5	[CLS] Great movie! 
Tuning-free Prompting	Frozen	-	-	GPT-3, AutoPrompt	Great movie! It was <mask> 
Fixed-prompt LM Tuning	Tuned	-	-	PET, LM-BFF	Great movie! It was <mask> 
Fixed-LM Prompt Tuning	Frozen	Yes	Yes	Prefix-Tuning, Prompt-Tuning	 Great movie! It was <mask> 
Prompt+LM Fine-tuning	Tuned	Yes	Yes	P-Tuning, PTR	 Great movie! It was <mask> 

Prompt Design Considerations:

5. Training Strategies

Strategy	LM Params	Prompt Parameters		Current Model	Advantage / Disadvantages
		Additional	Tuned		
Promptless Fine-tuning	Tuned	-	-	BERT, T5	+ Simplicity, no need for prompt design - LMs may overfit or not learn stably on smaller datasets
Tuning-free Prompting	Frozen	-	-	GPT-3, AutoPrompt	+ Efficiency, no parameter update. no catastrophic forgetting - Heavy engineering is necessary to achieve high accuracy
Fixed-prompt LM Tuning	Tuned	-	-	PET-, LM-BFF	+ Prompt more completely specify the task - LMs fine-tuned on one downstream task may not be effective on another one
Fixed-LM Prompt Tuning	Frozen	Yes	Yes	Prefix-Tuning, Prompt-Tuning	+ Efficiency, no parameter update. no catastrophic forgetting - Heavy engineering is necessary to achieve high accuracy
Prompt+LM Fine-tuning	Tuned	Yes	Yes	P-Tuning, PTR	+ Most expressive method, likely suitable for high-data settings - Requires training and storing all parameters of the models

Outline

- What is Prompt?
- Why Large Language Model Matters?
- How to Fine-tune the Language Model?
 - Apply to different tasks
- What are the design considerations for prompt-based methods?
 - **Example**
- Tools and Additional Resources
- Conclusion

Prompt Design Considerations:

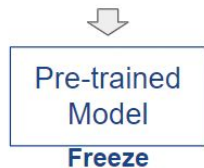
Example: Prompt Tuning

How to Fine-Tuning the Language Model?

The Power of Scale for Parameter-Efficient Prompt Tuning

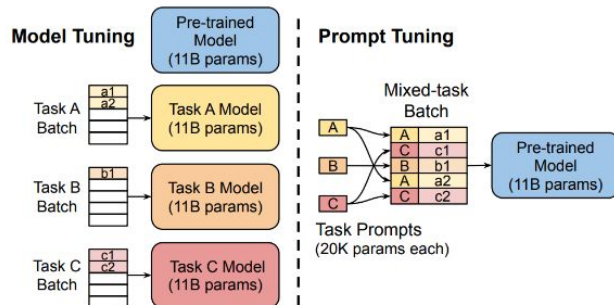
Prompt Tuning

<random vector> + **Input Text:**
A great movie !!!
Fine-Tune



Positive

- Fine-tuning with less parameters
- Storing a small task-specific prompt for each task



Pre-trained Model

- T5 (enc - dec)

Prompt Engineering (Template)

- Automated: Continuous

Answer Engineering (Verbalizer)

- Manual

Expanding the Paradigm
(Multi-Prompt Learning)


- None


Training Strategies

- Fixed-LM Prompt Tuning

Prompt Design Considerations:

Example: ChatGPT

 What is prompt in context of NLP?



In the context of natural language processing (NLP), a prompt is a piece of text used to initiate a task or query to a language model. It is a specific text input that is given to the language model to generate a response, and it can be a sentence, a paragraph, or a set of keywords that provide context and direction for the model to produce relevant output.

For instance, in the task of text generation, a prompt could be the beginning of a sentence, and the model will continue the sentence based on its understanding of language patterns and associations. In question answering, a prompt could be a question, and the model will generate an answer based on its knowledge of the topic.

Prompts can be manually crafted by humans or generated automatically using various techniques such as keyword extraction or clustering. They are a crucial component of NLP tasks and are used to guide the language model to produce specific output that aligns with the task at hand.

Pre-trained Model

- ChatGPT

Prompt Engineering (Template)

- Manual

Answer Engineering (Verbalizer)

- None

Expanding the Paradigm (Multi-Prompt Learning)

- None / Prompt Augmentation

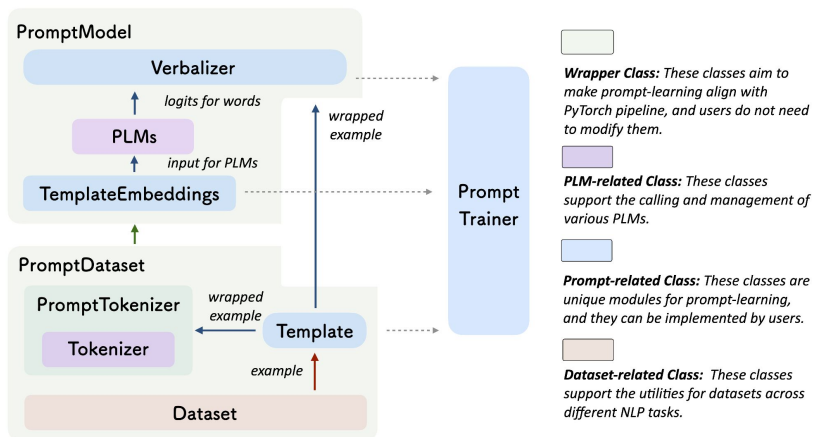
Training Strategies

- Tuning-free Prompting

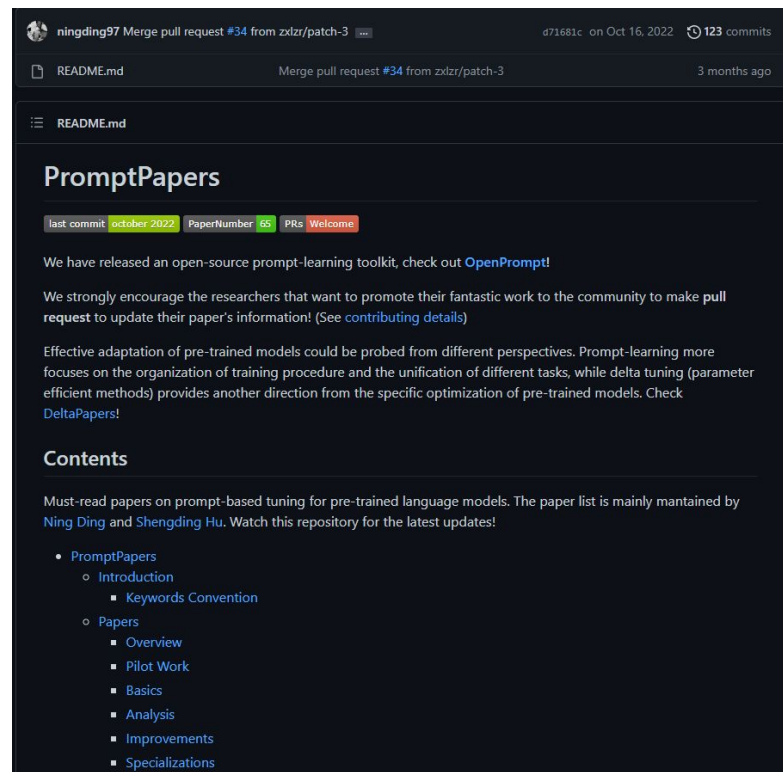
Outline

- What is Prompt?
- Why Large Language Model Matters?
- How to Fine-tune the Language Model?
 - Apply to different tasks
- What are the design considerations for prompt-based methods?
 - Example
- **Tools and Additional Resources**
- Conclusion

Tools and Additional Resources



<https://github.com/thunlp/OpenPrompt>



<https://github.com/thunlp/PromptPapers>

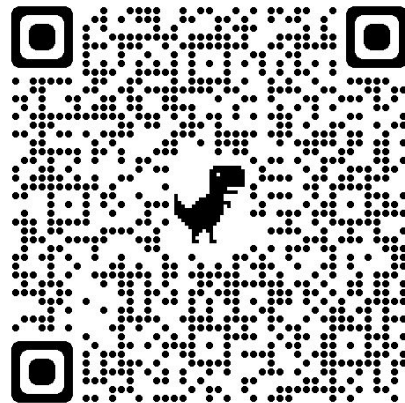
Coding Example

<https://colab.research.google.com/drive/1yruhfg4L9E3X29mEfSaC0lYTIKbb2jB2?usp=sharing>

<https://tinyurl.com/cuprompt>

Step:

1. Define a task
2. Obtain a PLM
3. Define a Template
4. Define a Verbalizer
5. Construct a PromptModel
7. Train and inference



Outline

- What is Prompt?
- Why Large Language Model Matters?
- How to Fine-tune the Language Model?
 - Apply to different tasks
- What are the design considerations for prompt-based methods?
 - Example
- Tools and Additional Resources
- **Conclusion**

Conclusion

What is Prompt in context of NLP?

- A piece of text used to guide a NLP model's generation of text or completion of a task

Why Large Language Model Matters?

- In some tasks, the tuning-free LLM achieves similar accuracy compared to fine-tune models

How to Fine-tune the Language Model?

- Transform input/task with template and verbalizer

What are the design considerations for prompt-based methods?

- 5: Pre-train model, template, verbalizer, multi-prompt learning, training strategies

Tools and Additional Resources

- Openprompt

References

- ❖ Dr Alan D. Thompson, **LifeArchitect.ai** (Month name/Year of creation). e.g. Dr Alan D. Thompson, LifeArchitect.ai (Mar/2023).
- ❖ T. Brown et al., “**Language Models are Few-Shot Learners**,” in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf
- ❖ N. Ding et al., “**OpenPrompt: An Open-source Framework for Prompt-learning**,” in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 105–113. doi: 10.18653/v1/2022.acl-demo.10.
- ❖ P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “**Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing**,” ACM Comput. Surv., vol. 55, no. 9, p. 195:1-195:35, Jan. 2023, doi: 10.1145/3560815.
- ❖ B. Lester, R. Al-Rfou, and N. Constant, “**The Power of Scale for Parameter-Efficient Prompt Tuning**,” in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. doi: 10.18653/v1/2021.emnlp-main.243.

Thank You