Outline

- **What is the "Prompt"?**
- **What is the general workflow of prompt-based methods?**
- **What are the design considerations for prompt-based methods?**
- **What (unique) advantages could prompt learning bring to us?**
- **How does prompt-based research progress currently?**

What is the “Prompt”?

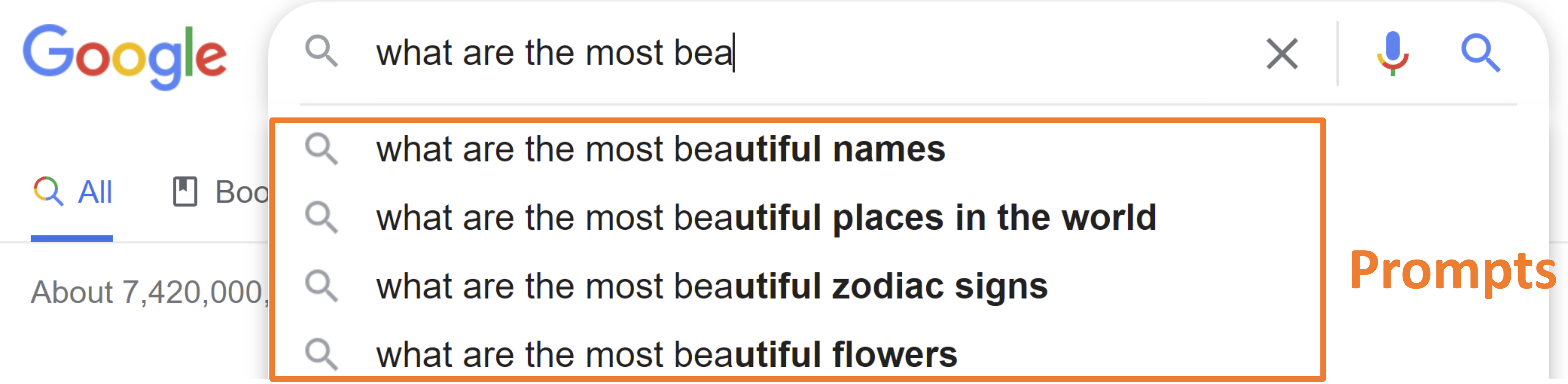
Prompt meaning prŏmpt

Words form: prompted promptest prompting prompts See word origin

The definition of a prompt is a cue given to someone to help him remember what to say, or is something that causes another event or action to occur. verb

An example of prompt is when you whisper a line to an actor who forgot what to say next.

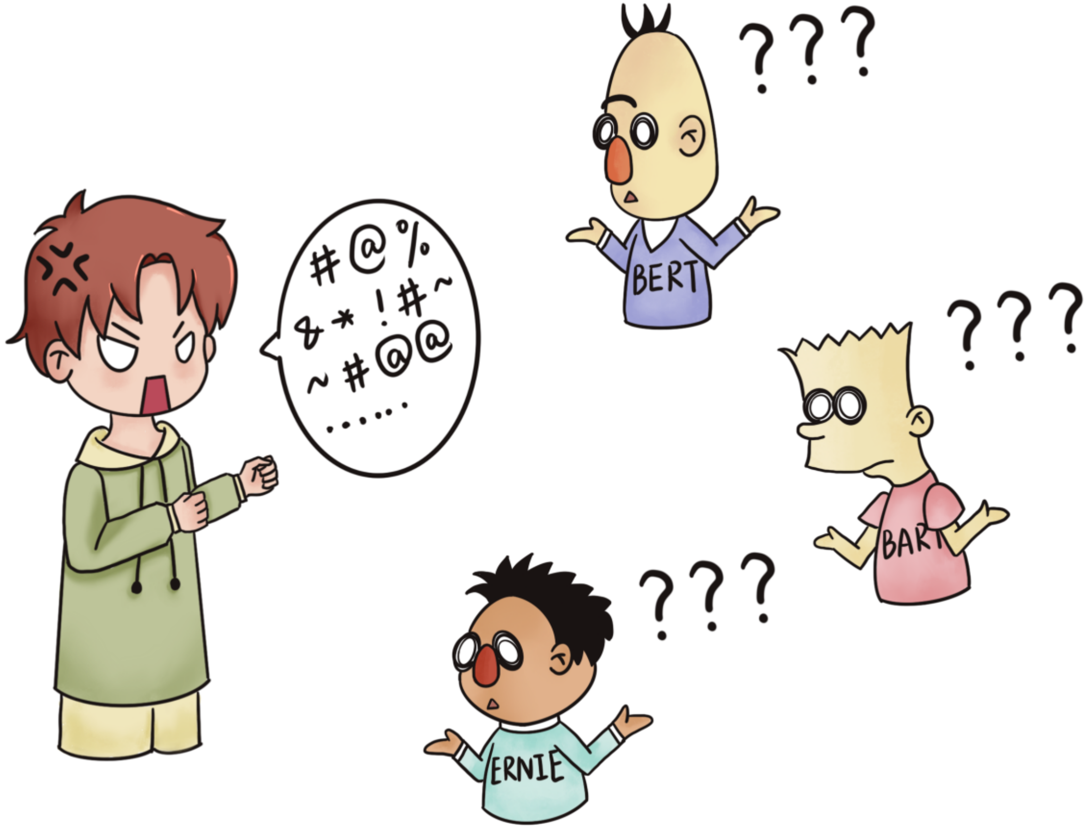
An example of prompt is an event that starts an argument.


Google
what are the most bea
All
Boo
About 7,420,000,
what are the most beautiful names
what are the most beautiful places in the world
what are the most beautiful zodiac signs
what are the most beautiful flowers
Prompts

What is the “prompt” in the context of NLP research?

An Intuitive Definition

- Prompt is a cue given to the **pre-trained language model** to allow it better understand **human**'s questions



More Technical Definition

- Prompt is the technique of making better use of the knowledge from the pre-trained model by adding additional texts to the input.

More Technical Definition

Purpose

- Prompt is the technique of making better use of the knowledge from the pre-trained model by adding additional texts to the input.

Method

task reformulation

What is the general workflow of prompt-based methods?

Workflow for Prompting Methods

- Prompt Construction
- Answer Construction
- Answer Prediction
- Answer-Label Mapping

Prompting for Sentiment Classification

- **Task Description**:
  - **Input**: sentence x;
  - **Output**: emotional polarity of it (i.e.,☺ v.s ☹).

Input: x = I love this movie.

Step 1: Prompt Construction

- Transform x into prompt x’ through following two steps:
  - Defining a **template** with two **slots**: [x] and [z];

Input: x = I love this movie.

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

Step 1: Prompt Construction

- Transform x into prompt x' through following two steps:
  
  - Defining a **template** with two **slots**: [x] and [z];

Require human effort

Input: x = I love this movie.

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

Step 1: Prompt Construction

- Transform x into prompt x’ through following two steps:

  - ☐ Defining a **template** with two **slots**: [x] and [z];
  - ☐ Instantiate slot [x] with input text.

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.

Step 2: Answer Construction

- Build a mapping function between answers and class labels.
  - ☺ -> fantastic
  - ☹ -> boring

label *answer*

Input: x = I love this movie.

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

Answer: {fantastic:☺, boring:☹}

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

Step 3: Answer Predicting

- Given a prompt， predict the answer [z].

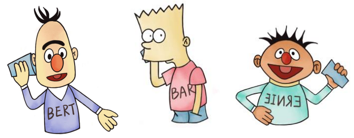
  - ☐ Choose a suitable pretrained language model;

Input: x = I love this movie.

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

Answer: {fantastic:☺, boring:☹}

Prompting: x’ = I love this movie. Overall, it was a [z] movie.



Which one?

Step 3: Answer Predicting

- Given a prompt， predict the answer [z]

- ☐ Choose a suitable pretrained language model;

- ☐ Fill in [z] as “fantastic”

Input: x = I love this movie.

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

Answer: {fantastic:☺, boring:☹}

Prompting: x’ = I love this movie. Overall, it was a [z] movie.

Predicting: x’ = I love this movie. Overall, it was a fantastic movie.

Step 4: Answer Mapping

- Mapping: Given an answer, map it into a class label.
  - fantastic => ☺

Input: x = I love this movie.

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

Answer: {fantastic:☺, boring:☹}

Prompting: x’ = I love this movie. Overall, it was a [z] movie.

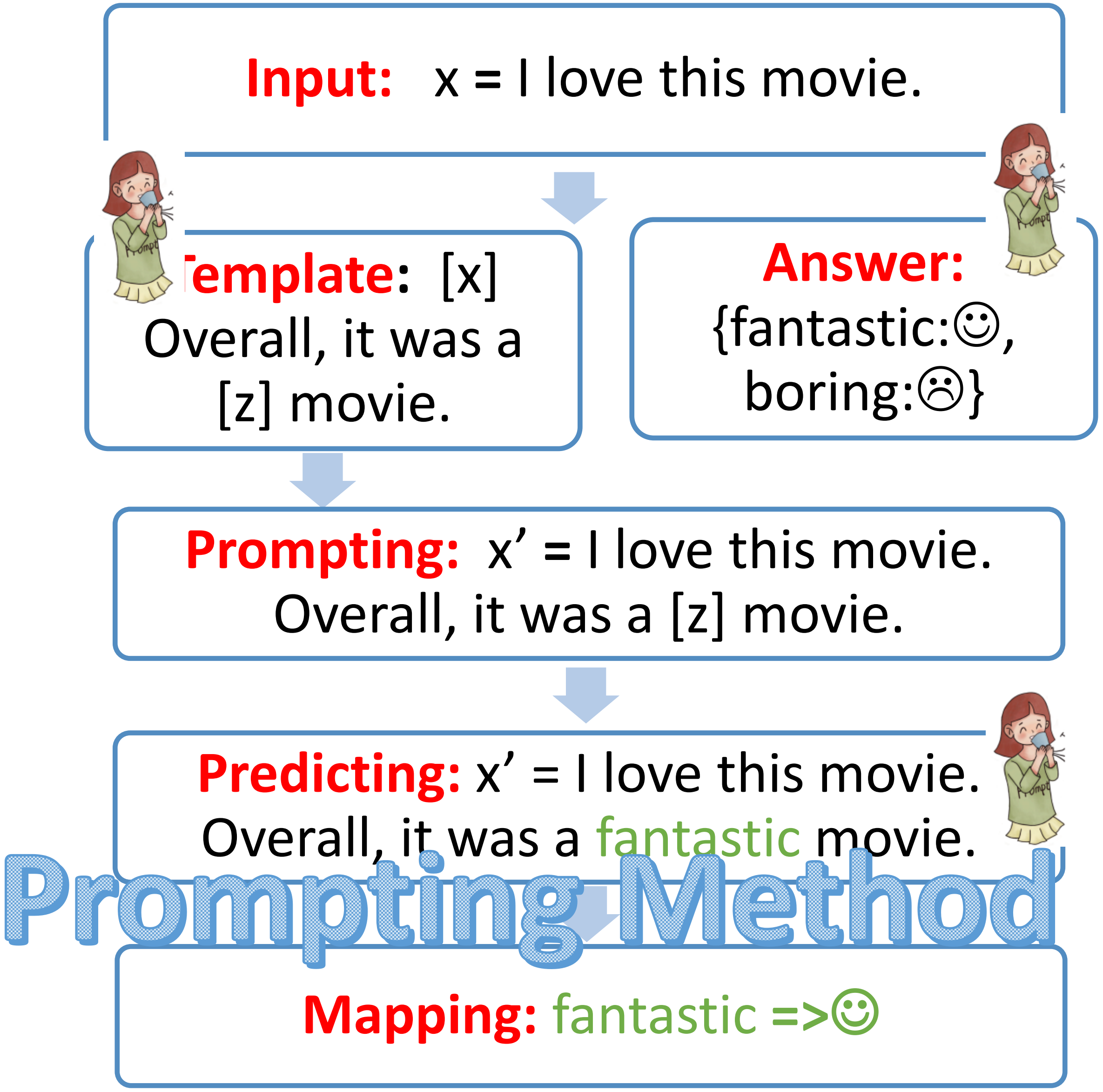
Predicting: x’ = I love this movie. Overall, it was a fantastic movie.

Mapping: fantastic =>☺

Summary

Terminology	Notation	Example
Input	x	I love this movie
Output (label)	y	☺ ☹
Template	-	[x] Overall, it was a [z] movie
Prompt	x’	I love this movie. Overall, it was a [z] movie
Answer	z	fantastic, boring

Rethinking Human Efforts in Prompt-based Methods



Rethinking Human Efforts in Prompt-based Methods

Input: x = I love this movie.

Predicting: ☺

Traditional Method

Input: x = I love this movie.

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

Answer: {fantastic:☺, boring:☹}

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

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

Prompting Method

Mapping: fantastic =>☺

What are the design considerations for prompt-based methods?

Design Considerations for Prompt-based Methods

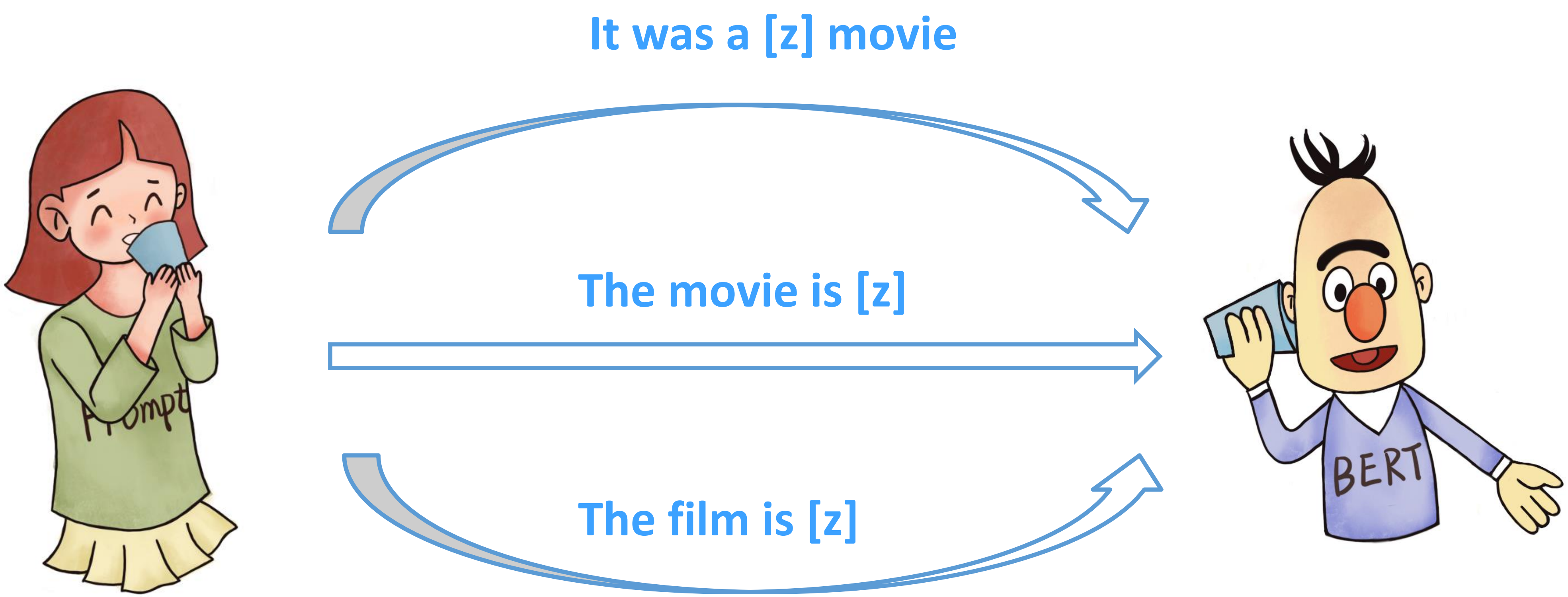
- Prompt Template Engineering
- Answer Engineering
- Pre-trained Model Choice
- Expanding the Paradigm
- Prompt-based Training Strategies

Design Considerations for Prompt-based Methods

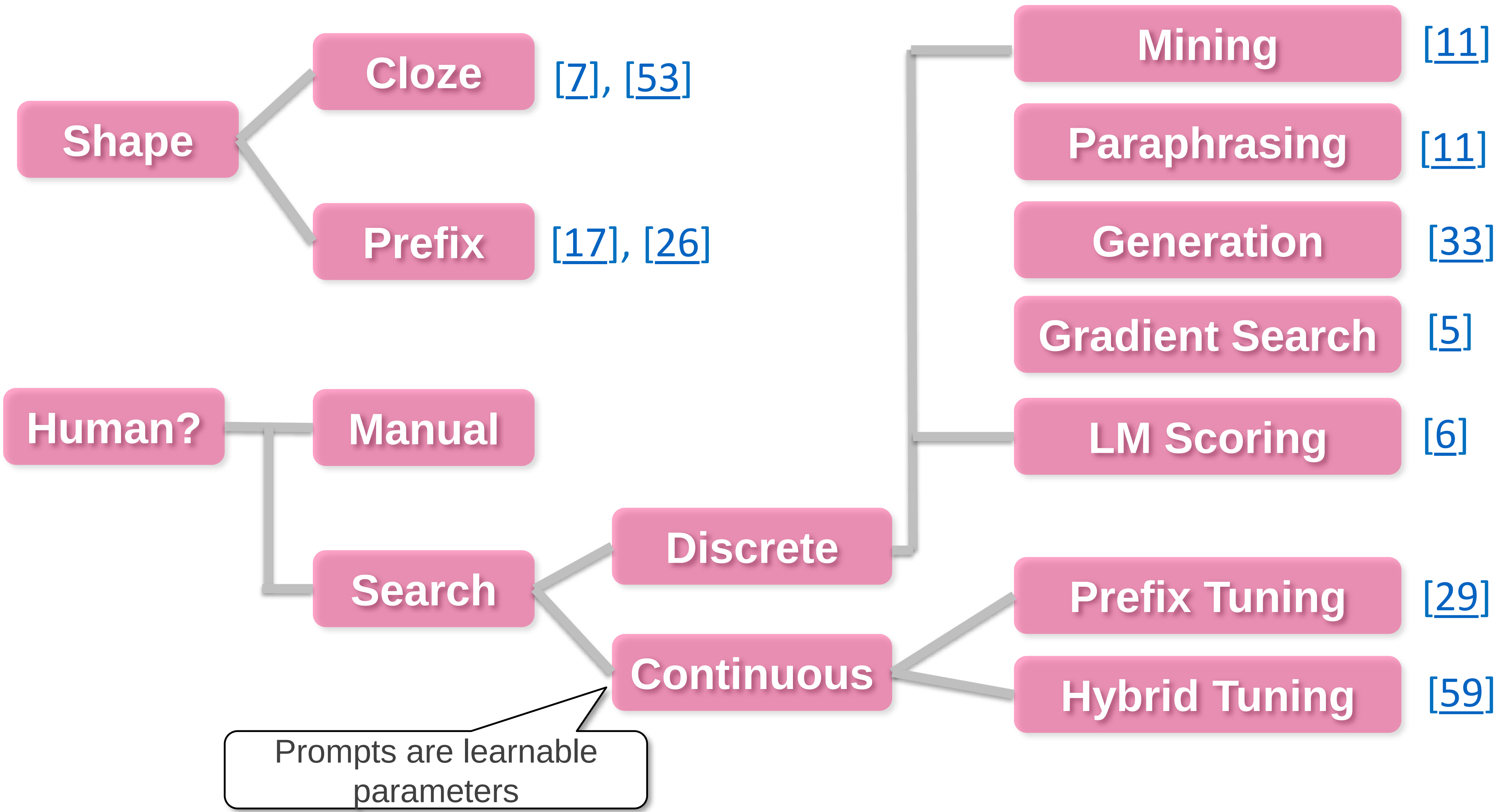
- **Prompt Template Engineering**
- Answer Engineering
- Pre-trained Model Choice
- Expanding the Paradigm
- Prompt-based Training Strategies

Prompt Template Engineering

- Research Question:
  - how to define appropriate prompt templates



Design Decision of Prompt Templates



One Example

I love this movie.

I love this movie.
Overall it was a **[z]** movie.

I love this movie.
Overall the movie is **[z]**

I love this movie.
[e][e] [e] [e]**[z]**

One Example

Prompt shape

I love this movie.

I love this movie.
Overall it was a **[z]** movie.

Cloze

I love this movie.
Overall the movie is **[z]**

Prefix

I love this movie.
[e][e] [e] [e]**[z]**

Prefix

One Example

Construction Way

I love this movie.

I love this movie.
Overall it was a [z] movie.

Cloze

Manual

I love this movie.
Overall the movie is [z]

Prefix

Search

Discrete

nonsense token

I love this movie.
[e][e] [e] [e][z]

Prefix

Search

Continuous

Design Considerations for Prompt-based Methods

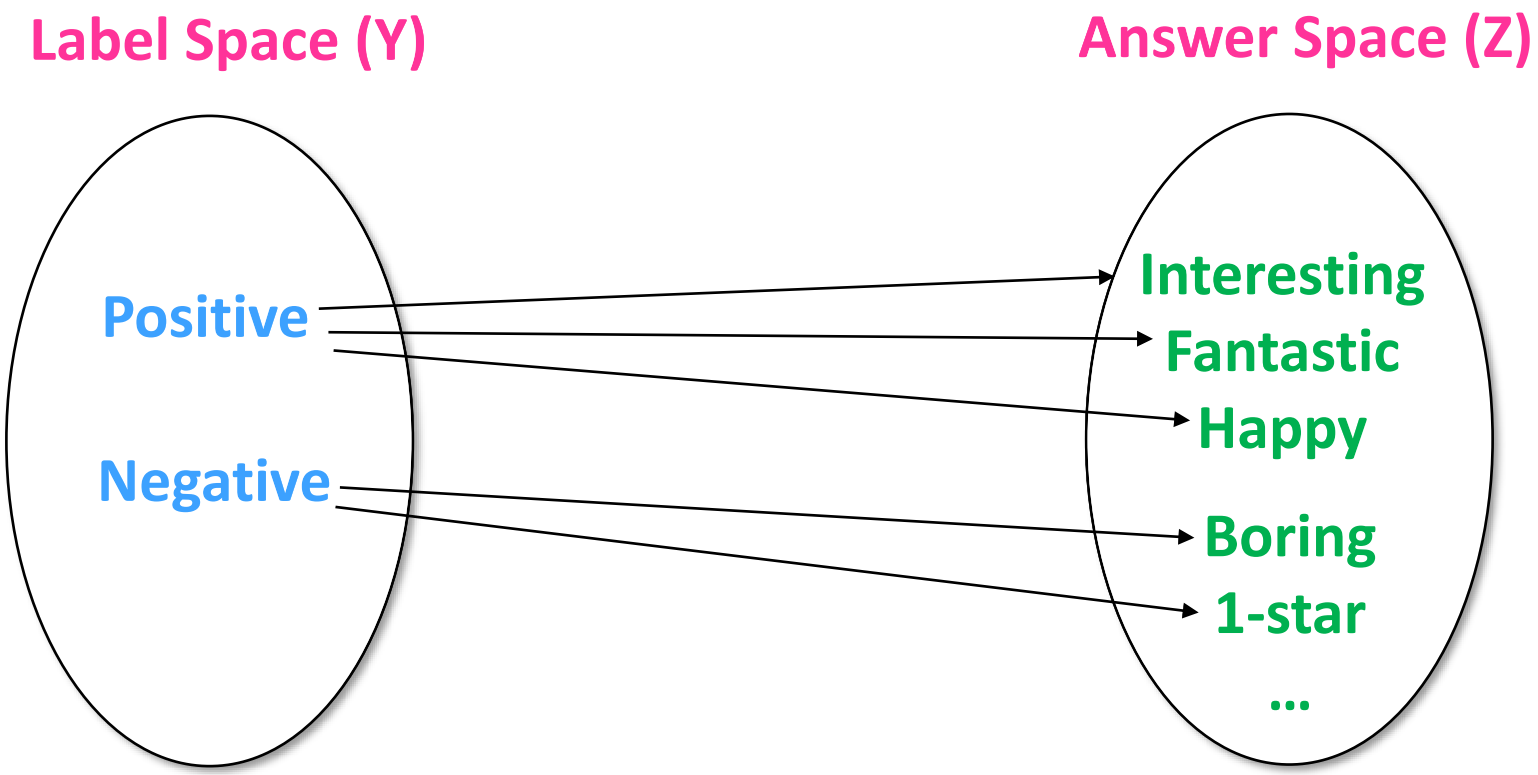
- Prompt Template Engineering
- **Answer Engineering**
- Pre-trained Model Choice
- Expanding the Paradigm
- Prompt-based Training Strategies

Answer Engineering

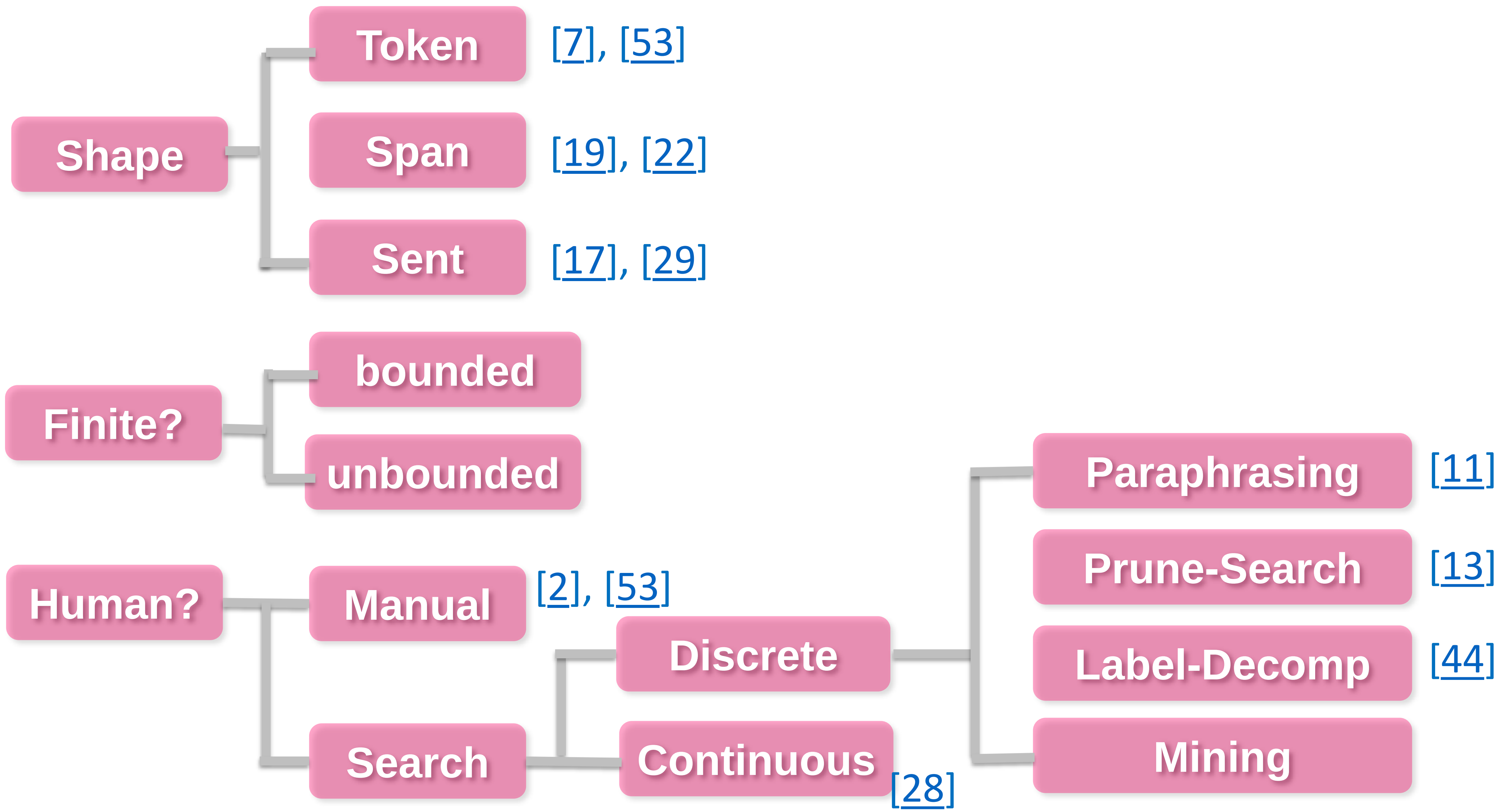
- Research Question:
  - ☐ Given a task (or a prompt), **how to define a suitable mapping function between label space and answer space?**

Answer Engineering

- Research Question:
  - Given a task (or a prompt), **how to define a suitable mapping function between label space and answer space?**



Design Decision of Prompt Answer Engineering



Task	Template	Answer		
			Token	
Sentiment Classification	[x] the movie is [z]	great fantastic boring	bounded	
			Manual	Search

Task	Template	Answer	
Sentiment Classification	[x] the movie is [z]	great fantastic boring	Token bounded Manual Search
Summarization	[x] in summary, [z]	The news...	Sentence unbounded Manual

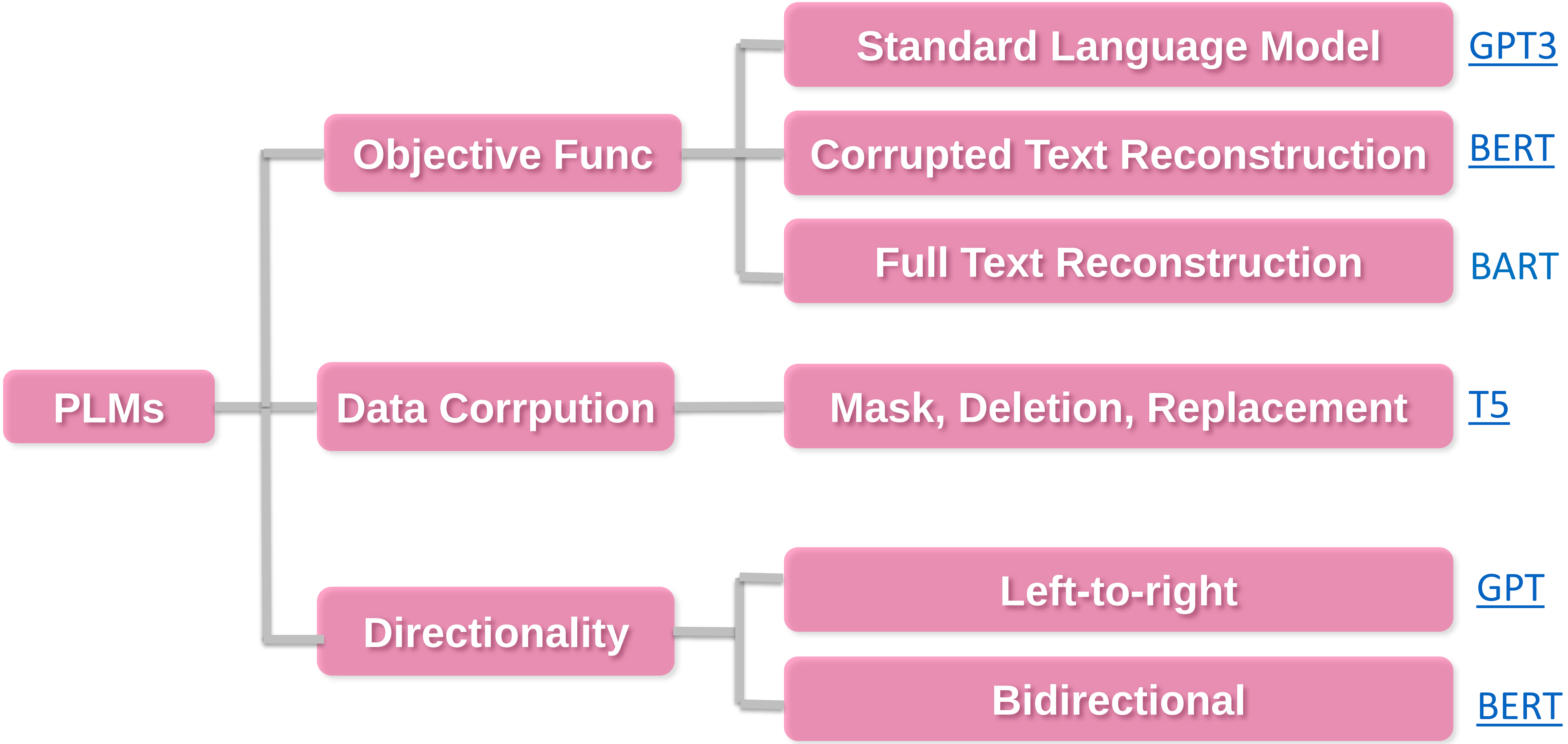
Design Considerations for Prompt-based Methods

- Prompt Template Engineering
- Answer Engineering
- **Pre-trained Model Choice**
- Expanding the Paradigm
- Prompt-based Training Strategies

Pre-trained Model Choice

- **Research Question**:
  - ☐ Given a task (or a prompt), **which pre-trained language model would be the most appropriate one?**

Design Decision of Pre-trained Models

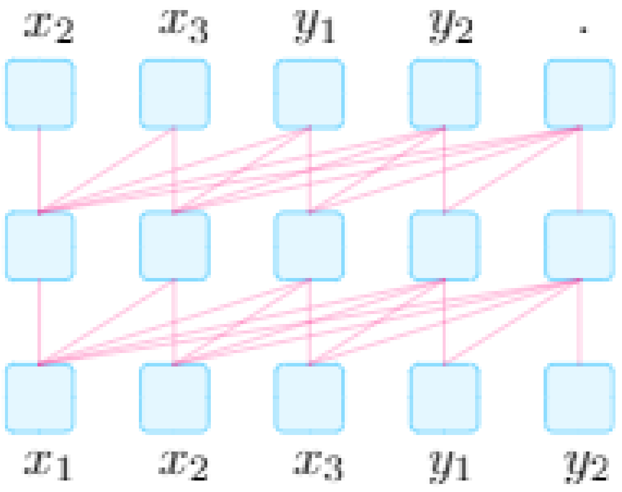


Design Decision of Pre-trained Models

- PLMs
  - Left-to-Right LM — GPT, GPT3
  - Masked LM — BERT, RoBERT
  - Encoder-decoder — BART, T5

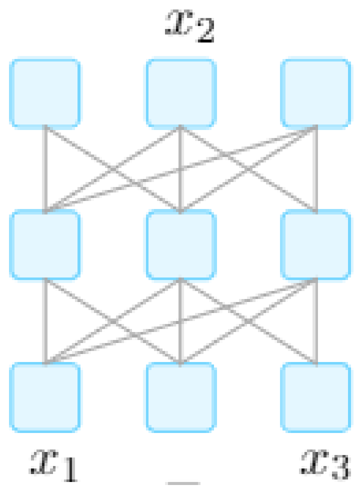
Left-to-right Language Model

- **Characteristics**
  - First proposed by Markov (1913)
  - Count-based-> Neural network-based
  - Specifically suitable to highly larger-scale LMs
- **Example**
  - GPT-1,GPT-2,GPT-3
- **Roles in Prompting Methods**
  - The earliest architecture chosen for prompting
  - Usually equipped with prefix prompt and the parameters of PLMs are fixed



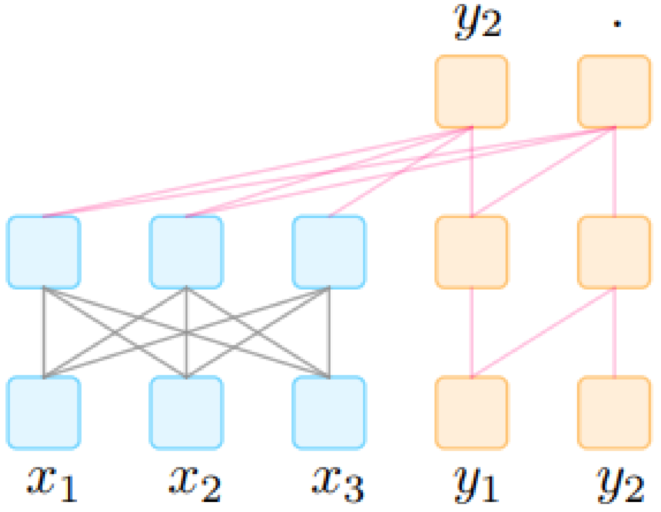
Masked Language Model

- **Characteristics**
  - An extension of left-to-right architecture
  - Unidirection -> bidirection prediction
  - Suitable for NLU tasks
- **Example**
  - BERT, ERNIE
- **Roles in Prompting Methods**
  - Usually combined with cloze prompt
  - Suitable for NLU tasks



Encoder-Decoder

- **Characteristics**
  - A denoised auto-encoder
  - Use two Transformers and two different mask mechanisms to handle text X and Y separately
- **Examples**
  - BART, T5
- **Roles in Prompting methods**
  - Text generation tasks or some tasks that can be formulated into a text generation problem



Design Considerations for Prompt-based Methods

- Prompt Template Engineering
- Answer Engineering
- Pre-trained Model Choice
- **Expanding the Paradigm**
- Prompt-based Training Strategies

Expanding the Paradigm

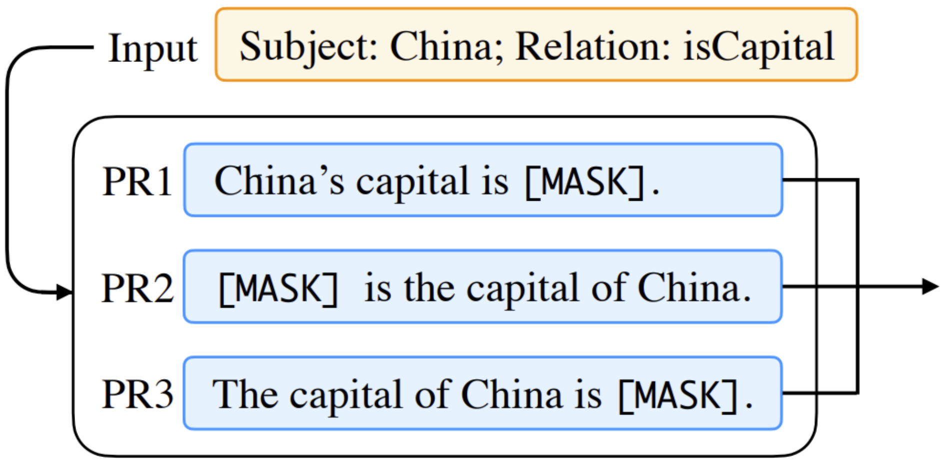
- Research Questions
  - How to extend the current prompting framework to support more NLP tasks?

Design Decision of Multiple Prompt Learning

- Multi-Prompt
  - Prompt Ensemble [11], [57]
  - Prompt Augmentation [47], [48]
  - Prompt Composition [52]
  - Prompt Decomposition [53]
  - Prompt Sharing

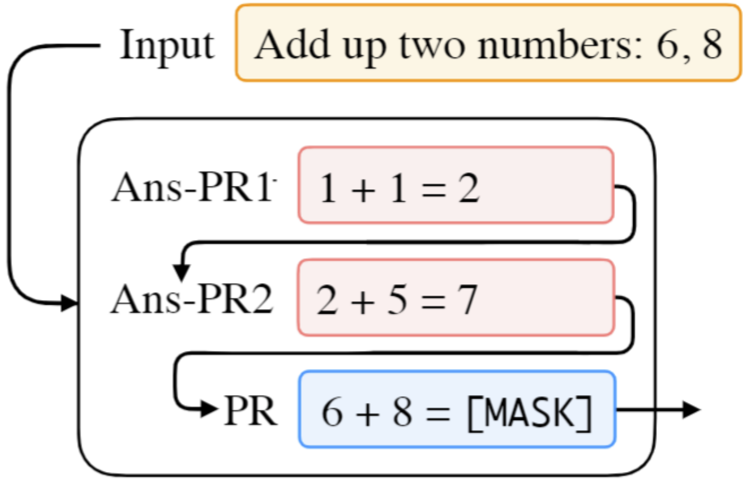
Prompt Ensembling

- Definition
  - using multiple unanswered prompts for an input at inference time to make predictions
- Advantages
  - Utilize complementary advantages
  - Alleviate the cost of prompt engineering
  - Stabilize performance on downstream tasks



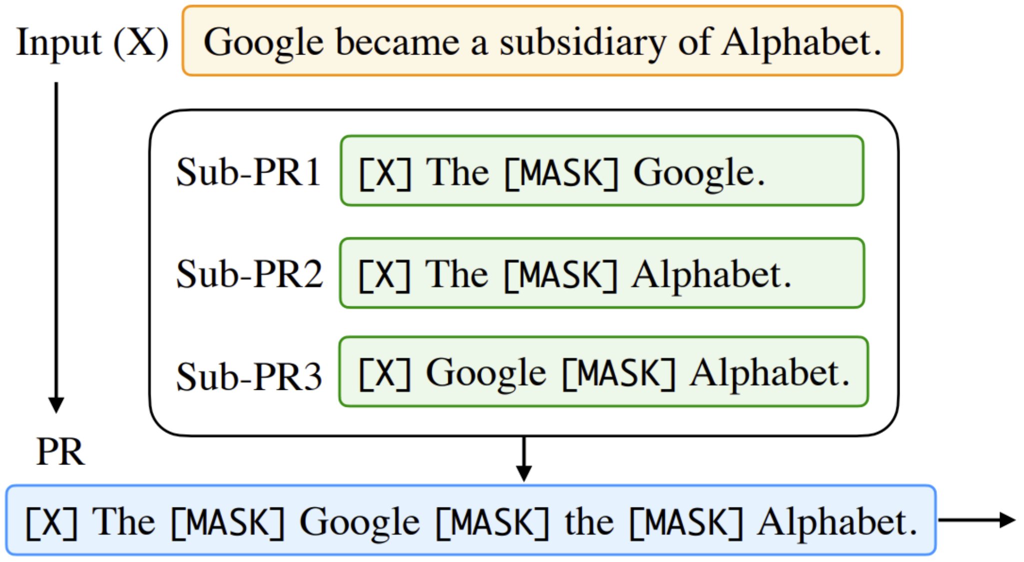
Prompt Augmentation

- Definition
  - Help the model answer the prompt with additional answered prompts
- Advantage
  - make use of the small amount of information that has been annotated
- Core step
  - Selection of answered prompts
  - Ordering of answered prompts



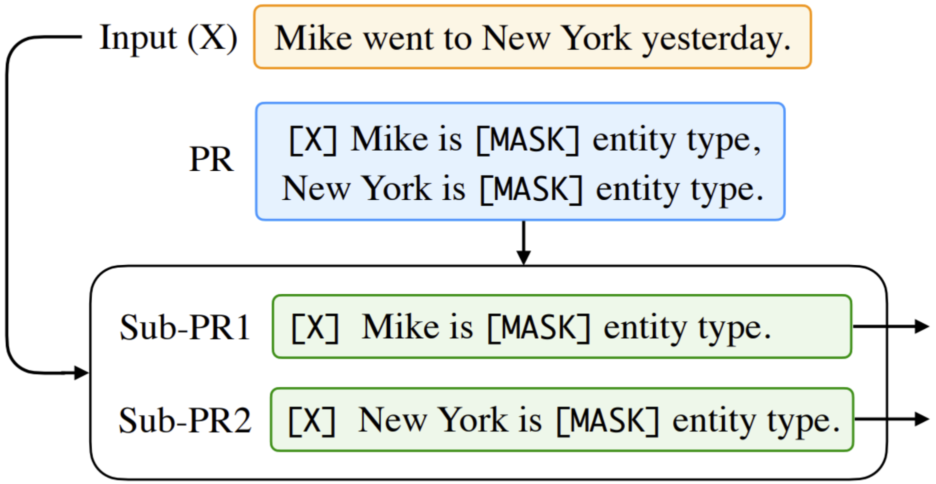
Prompt Composition

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



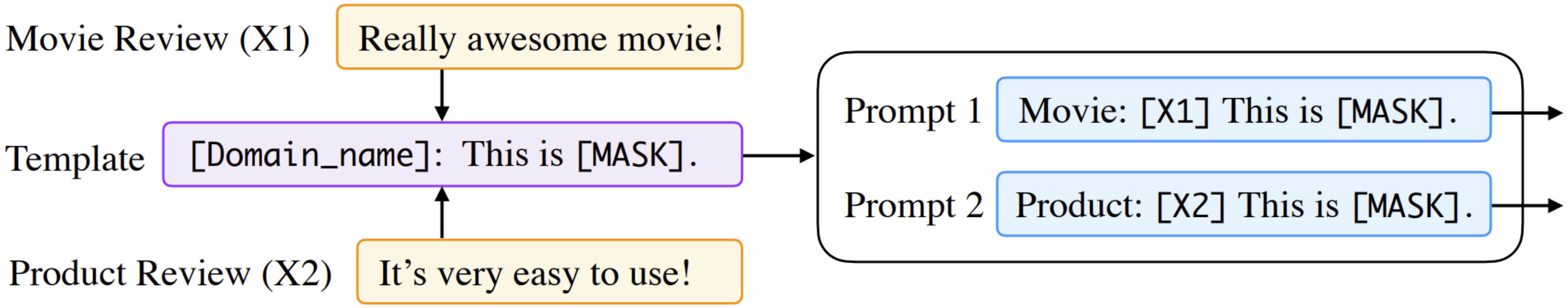
Prompt Decomposition

- Definition
  - 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 Sharing

- Definition
  - When prompting method is applied to multiple tasks, domains or languages , prompts can be shared cross different tasks.
- Advantage
  - Task- or language invariant information can be captured through prompting.



Design Considerations for Prompt-based Methods

- Prompt Template Engineering
- Answer Engineering
- Pre-trained Model Choice
- Expanding the Paradigm
- **Prompt-based Training Strategies**

Prompt-based Training Strategies

- Data Perspective
  - Zero-shot: without any explicit training of the LM for the down-stream task
  - Few-shot: few training (e.g., 100) samples of downstream tasks
  - Full-data: lots of training samples (e.g., 10K) of downstream tasks

Parameter Perspective

Pre-trained LMs

tuned

frozen

Prompts

no

without

frozen

Tuned

No prompts

No parameters
(discrete
prompts)

Cases of Parameter Updating

Pre-trained LMs

tuned frozen

Prompts

no without frozen Tuned

Promptless Fine-tuning

Example: BERT for text classification

Cases of Parameter Updating

Pre-trained LMs

tuned frozen

Prompts

no **without** frozen Tuned

Fixed-prompt Tuning

Example: BERT + Discrete Prompt for text classification

Cases of Parameter Updating

Pre-trained LMs

tuned frozen

Prompts

no without **frozen** Tuned

Fixed-prompt Tuning

Example: BERT + Transferred Continuous Prompt for text classification

Cases of Parameter Updating

Pre-trained LMs

tuned frozen

Prompts

no without frozen **Tuned**

Prompt+LM Fine-tuning

Example: BERT + Continuous Prompt for text classification

Cases of Parameter Updating

Pre-trained LMs

tuned

frozen

Prompts

no

without

frozen

Tuned

Adapter Tuning

Example: BERT + Adapter for text classification

Cases of Parameter Updating

Pre-trained LMs

tuned **frozen**

Prompts

no **without** frozen Tuned

Tuning-free Prompting

Example: GPT3 + Discrete Prompts for Machine Translation

Cases of Parameter Updating

Pre-trained LMs

tuned **frozen**

Prompts

no without **frozen** Tuned

Tuning-free Prompting

Example: GPT3 + Continuous Prompts for Machine Translation

Cases of Parameter Updating

Pre-trained LMs

tuned **frozen**

Prompts

no without frozen **Tuned**

Fixed-LM Prompt Tuning

Example: BART + Continuous Prompts for Machine Translation

Too many, difficult to select?

Promptless Fine-tuning

Fixed-prompt Tuning

Prompt+LM Fine-tuning

Adapter Tuning

Tuning-free Prompting

Fixed-LM Prompt Tuning

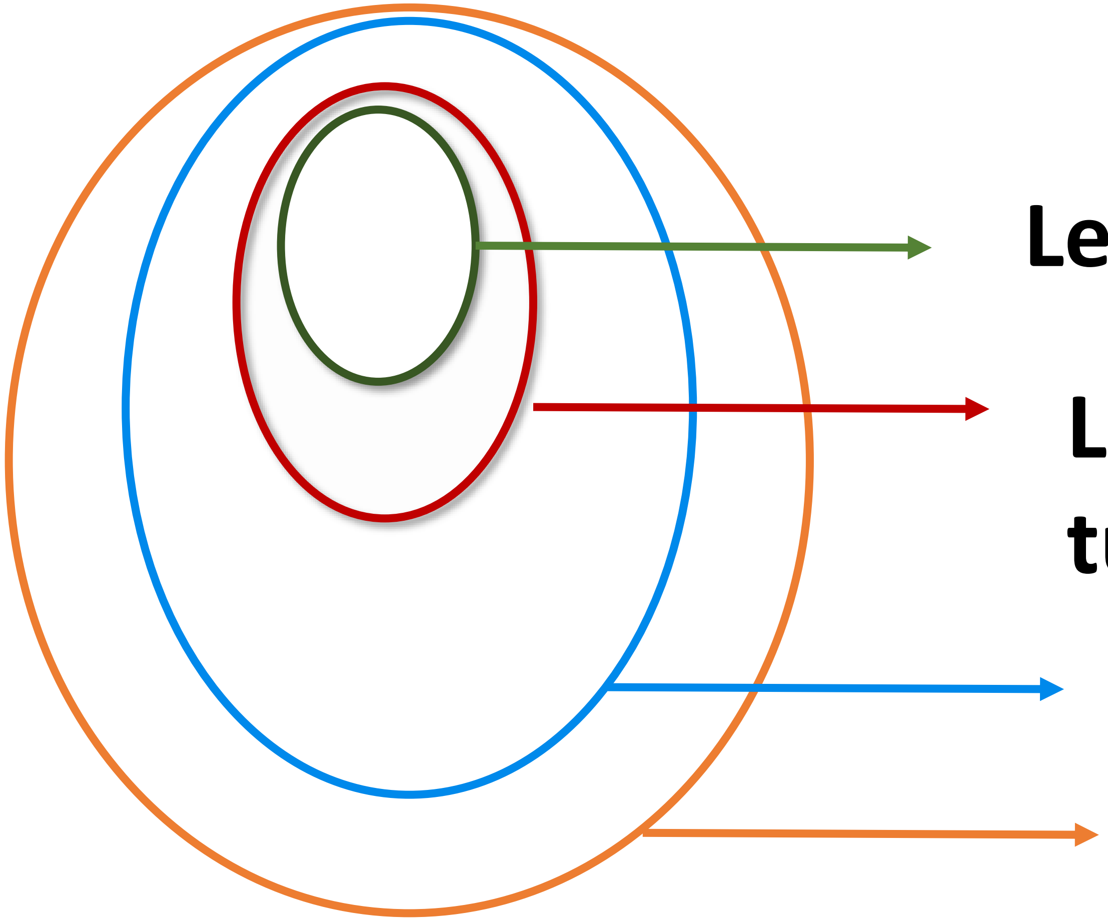
If you have a highly large left-to-right pre-trained language model (e.g., GPT3)

If you have few training samples?

If you have lots of training samples?

What (unique) advantages could prompt learning bring to us?

Four levels of vision



Level 1: Prompt learning

Level 2: Prompt learning v.s Fine-tuning

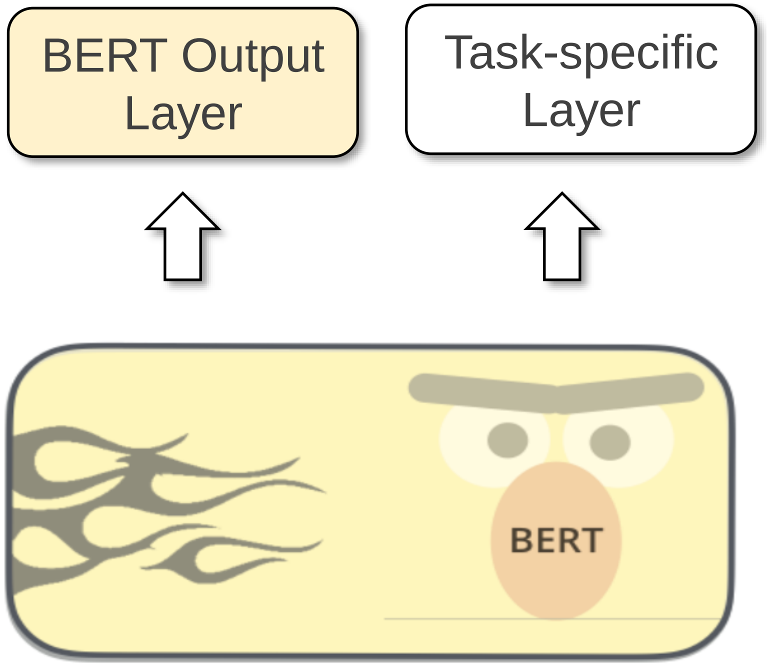
Level 3: Modern NLP history

Level 4: Beyond NLP

Level-1: Make All NLP Tasks as a Language Modeling Problem

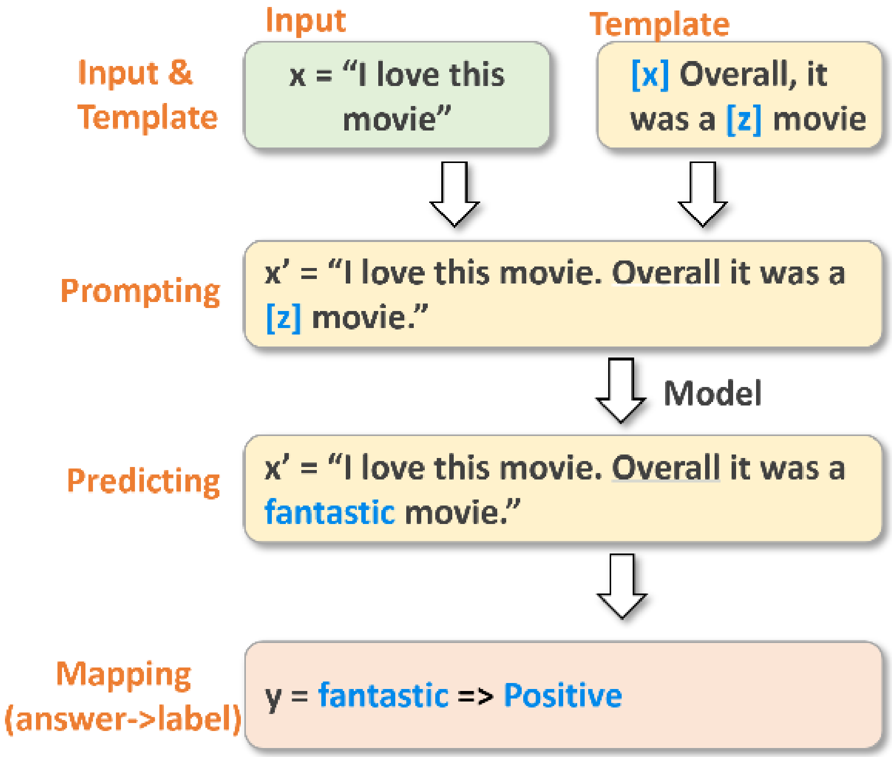
Level-1: Make All NLP Tasks as a Language Modeling Problem

- Pretrained Language models can be better utilized



Level-1: Make All NLP Tasks as a Language Modeling Problem

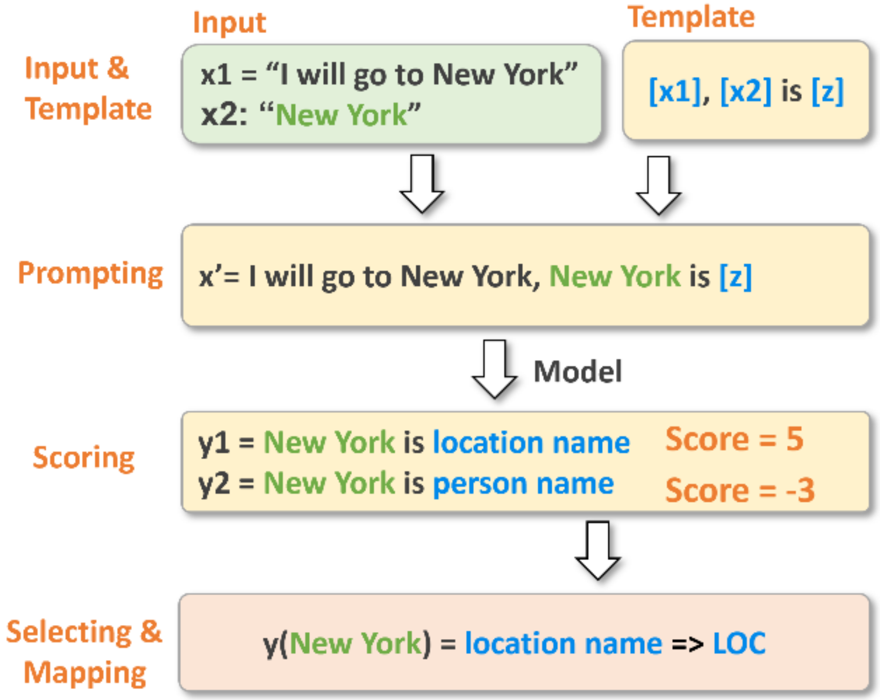
- Pretrained Language models can be fully utilized
- (Almost) all NLP tasks can be handled zero-shotly



Sentiment Classification

Level-1: Make All NLP Tasks as a Language Modeling Problem

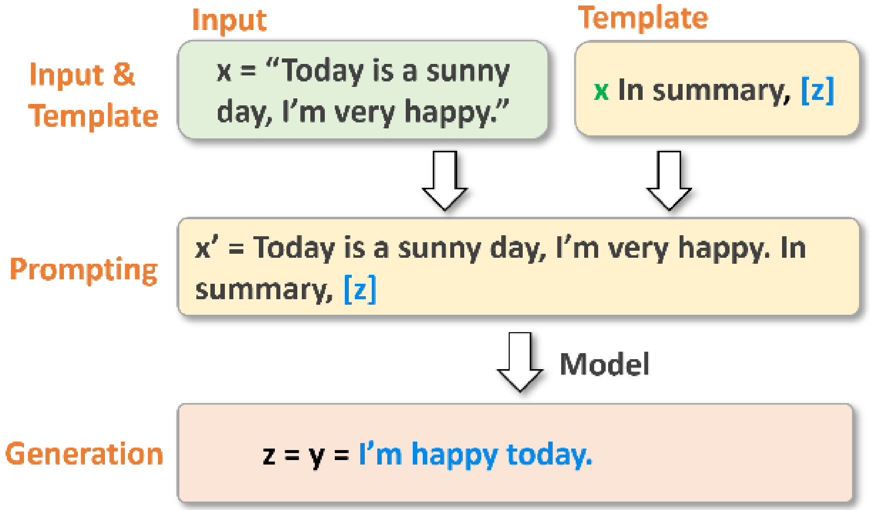
- Pretrained Language models can be fully utilized
- (Almost) all NLP tasks can be handled zero-shotly



Named Entity Recognition

Level-1: Make All NLP Tasks as a Language Modeling Problem

- Pretrained Language models can be fully utilized
- (Almost) all NLP tasks can be handled zero-shotly



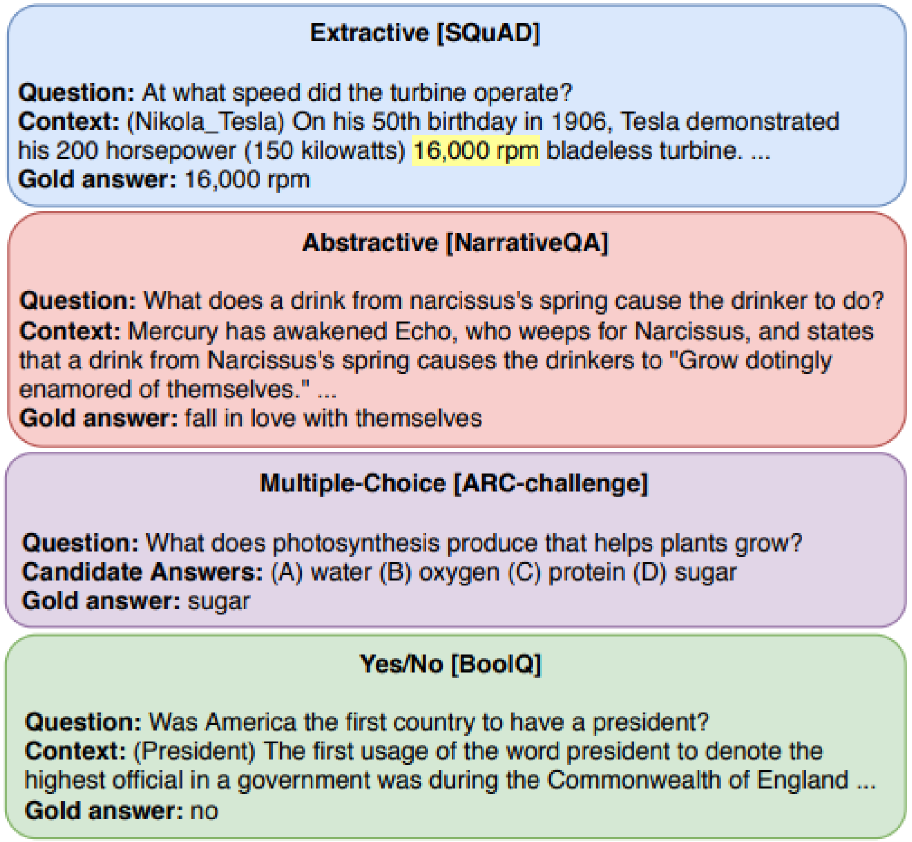
Text Summarization

Level-1: Make All NLP Tasks as a Language Modeling Problem

- Pretrained Language models can be fully utilized
- (Almost) all NLP tasks can be handled zero-shotly
- Better few-shot performance

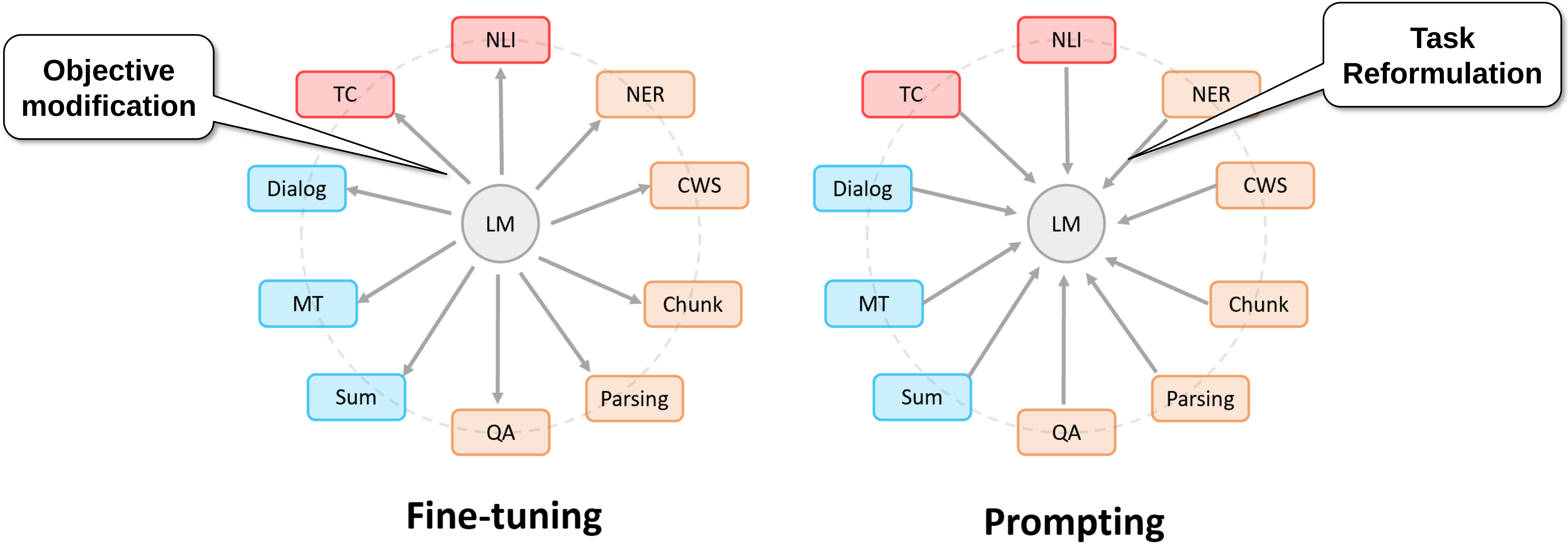
Level-1: (Almost) All NLP Tasks as Language Modeling

- Pretrained Language models can be fully utilized
- (Almost) all NLP tasks can be handled zero shotly
- Better few-shot performance
- Make different tasks methodologically-connected available



Unified QA (Daniel et al 2020)
Different QA tasks are trained using one model

Level-2: Reverse Thinking



Level-3: Reveal a "secret" about NLP development

- Prompting methods let out a secret how the technique of modern NLP progress

Four Paradigms in Modern NLP

- Feature Engineering
- Architecture Engineering
- Objective Engineering
- Prompt Engineering

Four Paradigms in Modern NLP

- Feature Engineering
- Architecture Engineering
- Objective Engineering
- Prompt Engineering

- **Paradigm**： Fully Supervised Learning (Non-neural Network)
- **Date**： Before 2013
- **Characteristic**： Traditional machine learning model is mainly used, which requires manual feature definition of input text
- **Typical Work**：
  - CRF (Conditional Random Field)

Four Paradigms in Modern NLP

- Feature Engineering
- Architecture Engineering
- Objective Engineering
- Prompt Engineering

- **Paradigm**： Fully Supervised Learning (Neural Network)
- **Date**： 2013 - 2018
- **Characteristic**：
  - Rely on neural networks
  - Do not need to manually define features, but should explore the network structure （e.g.：LSTM v.s CNN）
- **Typical Work**：
  - CNN for Text Classification

Four Paradigms in Modern NLP

- Feature Engineering
- Architecture Engineering
- Objective Engineering
- Prompt Engineering

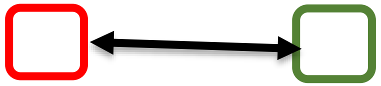
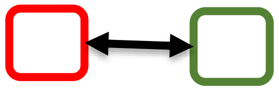
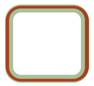
- **Paradigm**： Pre-train, Fine-tune
- **Date**： 2018-Now
- **Characteristic**：
  - context-dependent PLMs
  - Need to pay attention to the definition and selection of objective functions
- **Typical Work**： BERT

Four Paradigms in Modern NLP

- Feature Engineering
- Architecture Engineering
- Objective Engineering
- Prompt Engineering

- **Paradigm**： Pre-train, Prompt, Predict
- **Date**： 2019-Now
- **Characteristic**：
  - NLP tasks are modeled entirely by relying on PLMs
  - More efforts on prompt design
- **Typical Work:** GPT3

PLMs and Downstream Tasks are Getting Closer and Closer

Stages	Downstream Tasks Pre-trained LMs	Reasons
Traditional machine learning		No pre-training language model
Neural network methods enhanced by word2vec		The pre-trained language model plays the role of initializing the input text signal
The fine-tune method represented by BERT		The pre-trained language model is **responsible for extracting** high-level features from the input text
The prompt approach represented by GPT3		Pre-training language models **take on more responsibilities**: feature extraction, result prediction

Secret let out from Prompt-based Learning

The history of modern natural language processing is essentially (probably) a history of changes in the relationship between downstream tasks and pre-trained language models (PLMs).

Closer

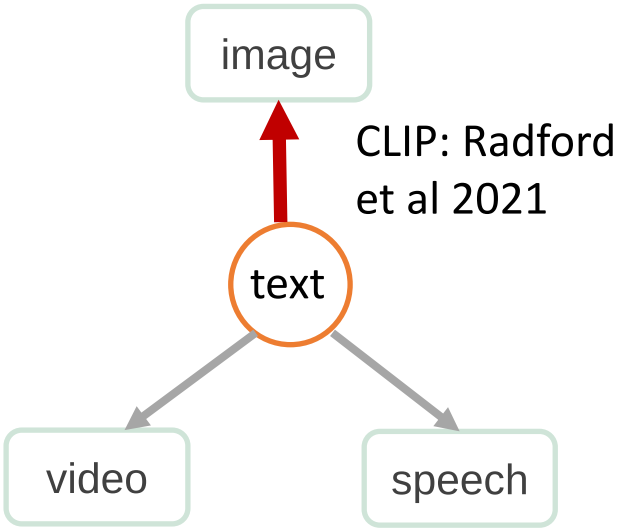
Downstream
Tasks

Pre-trained
Language Models

(1) use pre-trained language models
(2) use a better pre-trained language model
(3) better use a pre-trained language model

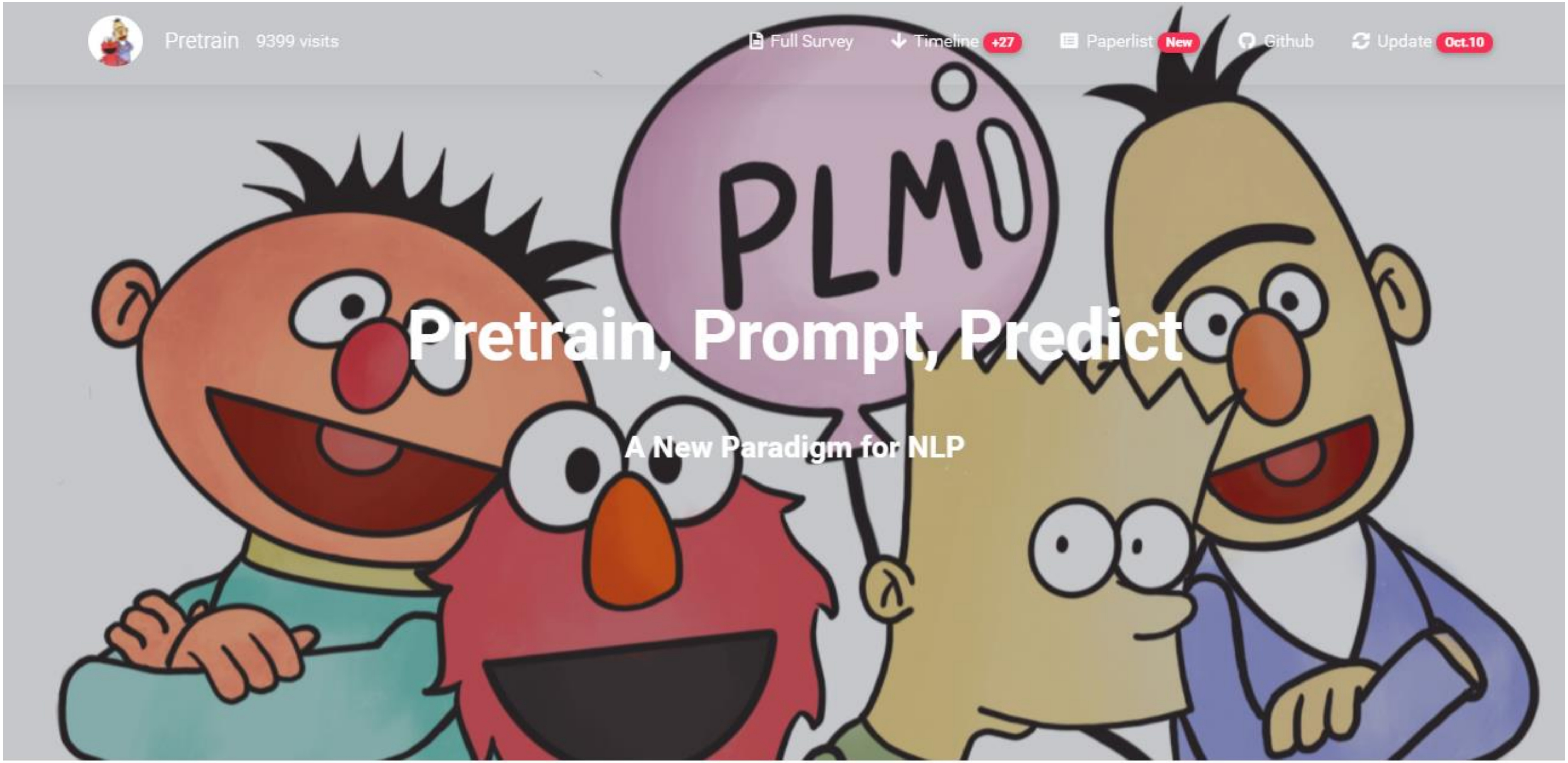
Level-4: Beyond NLP

- Prompting methods make
  - more modalities of signals (e.g. image) connected using natural language as relay node
- New view for human to **interact** with data in the world



How does prompt-based research progress currently?

Website Resource for Prompt-based Research



Website Resource for Prompt-based Research

- **Timeline**
- Paperlist

Timeline of Prompt Learning

- Revisiting Self-Training for Few-Shot Learning of Language Model — 04 October, 2021
  Prompt-fix LM Tuning
- Towards Zero-Label Language Learning — 19 September, 2021
  Tuning-free Prompting
- SentiPrompt: Sentiment Knowledge Enhanced Prompt-Tuning for Aspect-Based Sentiment Analysis — 17 September, 2021
  Fixed-prompt LM Tuning
- Reframing Instructional Prompts to GPTk's Language — 16 September, 2021
  Tuning-free Prompting
- Language Models are Few-shot Multilingual Learners — 16 September, 2021
  Tuning-free Prompting
- Can Language Models be Biomedical Knowledge Bases? — 15 September, 2021
  Tuning-free Prompting; Fixed-LM Prompt Tuning
- Dialogue State Tracking with a Language Model using Schema-Driven Prompting — 15 September, 2021
  Fixed-prompt LM Tuning
- Exploring Prompt-based Few-shot Learning for Grounded Dialog Generation — 14 September, 2021
  Fixed-Prompt LM Tuning; Fixed-LM Prompt Tuning
- Eliciting Knowledge from Language Models for Event Extraction — 11 September, 2021
  Fixed-Prompt LM Tuning
- An Empirical Study of GPT-3 for Few-Shot Knowledge-Based VQA — 10 September, 2021
  Tuning-free Prompting

Website Resource for Prompt-based Research

- Timeline
- **Paperlist**

ExplainaBoard - Prompt-based Learning

Full Survey can be obtained here.

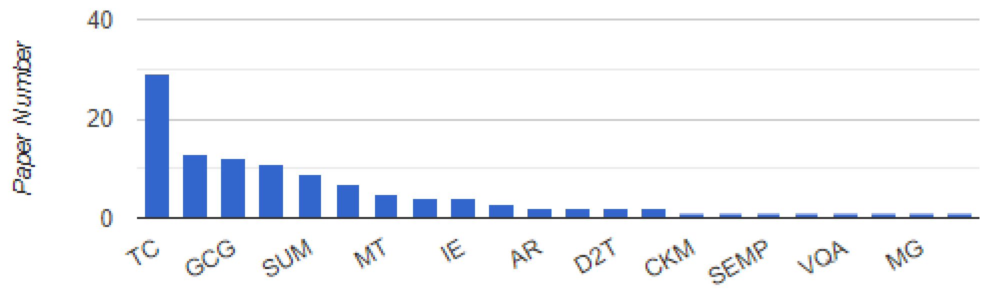
Year	☐ 2018 ☐ 2019 ☐ 2020 ☐ 2021 + more
Task	☐ AR ☐ CKM ☐ CR ☐ CodeGen ☐ D2T ☐ EVALG ☐ FP + more
Pretrained LMs	☐ ALBERT ☐ BART ☐ BERT ☐ CPM-2 ☐ CTRL ☐ Conv ☐ ELMo + more
Settings	☐ Few ☐ Full ☐ Zero + more
Prompt Mining	☐ automated ☐ hand-crafted + more

TREND ANALYSIS

Search:

	Year	Conf	Title	Task	PLMs	Citation	Bib
+	2021	NAACL	It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners Timo Schick,Hinrich Schütze	TC	ALBERT	58	Bib
+	2021	ICLR	A Mathematical Exploration of Why Language Models Help Solve Downstream Tasks Nikunj Saunshi,Sadhika Malladi,Sanjeev Arora	Theory	GPT-2	5	Bib
+	2021	ACL	Making Pre-trained Language Models Better Few-shot Learners Tianyu Gao,Adam Fisch,Danqi Chen	TC	RoBERTa	38	Bib

The number of papers over different tasks



Summary of Prompt-based Research

- How to apply prompting methods to diverse NLP tasks
- Tuning Strategy
- Non-NLP Tasks
- Annotation
- Pre-training
  - New pretraining framework

Summary of this talk

What is the "Prompt"?

- tool for human – PLM communication
- technique of making better use of pre-trained model by task reformulation

What is the general workflow?

- Prompt Construction
- Answer Construction
- Answer Prediction
- Answer-Label Mapping

What are the design considerations?

- Prompt Template Engineering
- Answer Engineering
- Pre-trained Model Choice
- Expanding the Paradigm
- Prompt-based Training Strategies

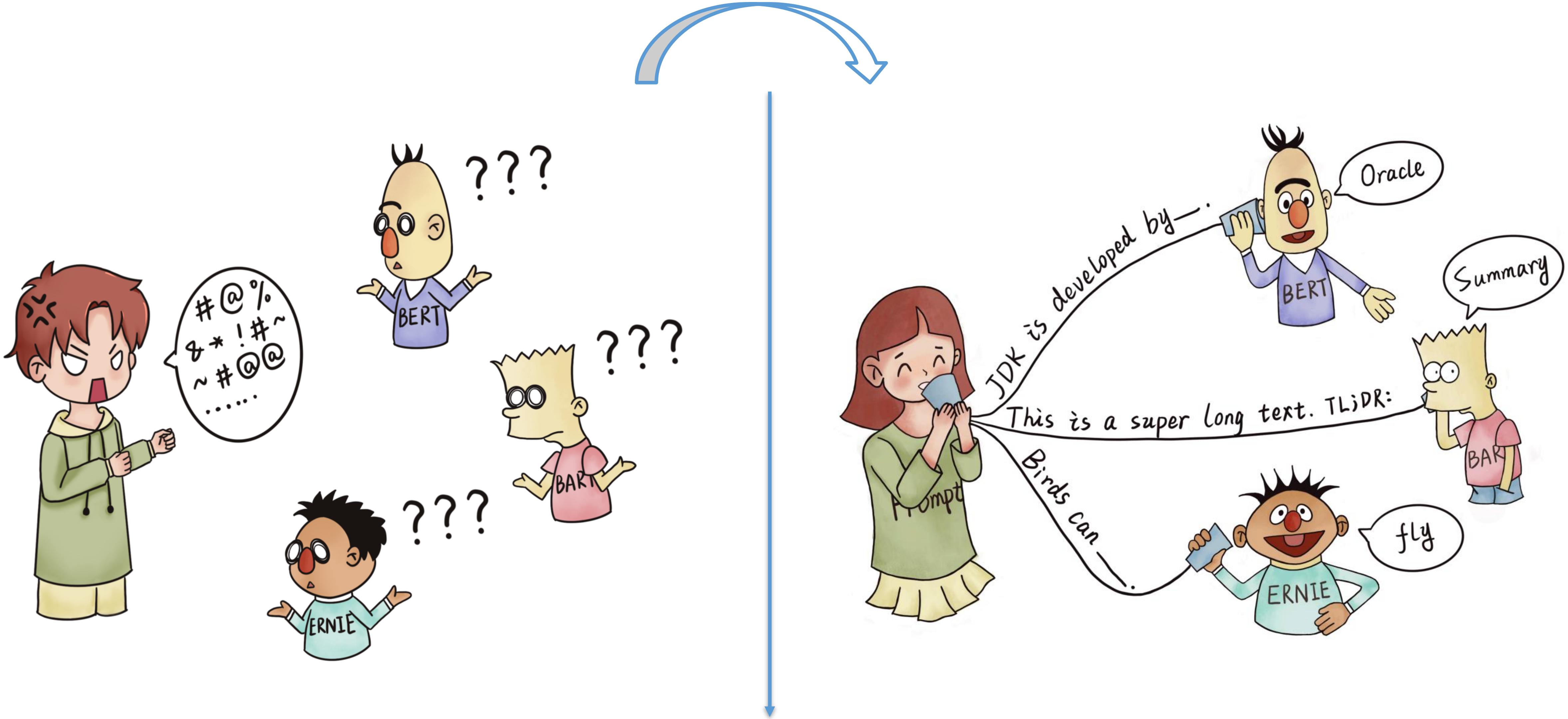
What (unique) advantages?

- Level-1: Within Prompt Learning
- Level-2: Prompt Learning v.s. Fine-tuning
- Level-3: Modern NLP History
- Level-4: Beyond NLP

How does prompt-based research progress?

- More diverse NLP tasks
- Tuning Strategy
- Non-NLP Tasks
- Annotation
- Pre-training

Hopefully


#@%
&*!#~
~#@@
......
???
BERT
???
BART
???
ERNIE
JDK is developed by—.
Oracle
BERT
This is a super long text. TL;DR:
Summary
BAR
Birds can—.
fly
ERNIE
Prompt