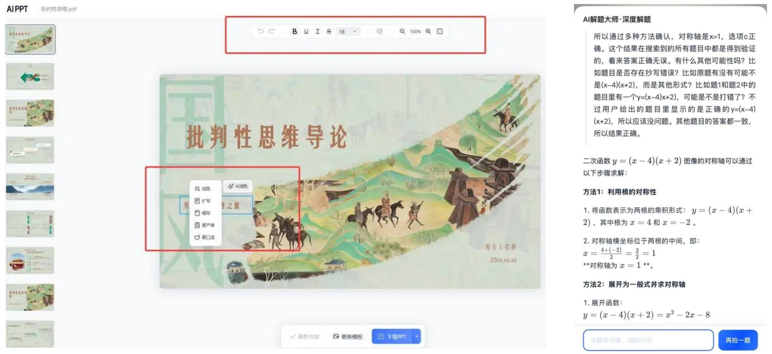
一周AI大事（2025.03.14～2025.03.20）

夸克上线超级Agent，鸣枪起跑!



国产机器人PM01跳斧头帮舞蹈

字节发布DAPO强化学习算法

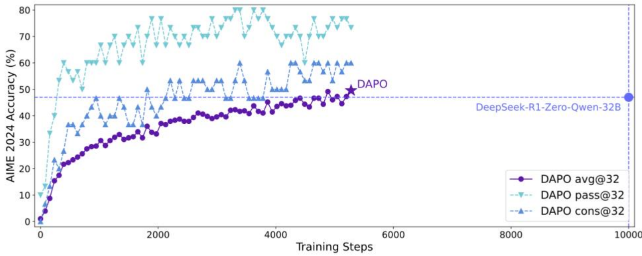


Figure 1 AIME 2024 scores of **DAPO** on the Qwen2.5-32B base model, outperforming the previous SoTA DeepSeek-R1-Zero-Qwen-32B using 50% training steps.

百度发布文心大模型X1和文心大模型4.5

OpenAI后训练负责人、研究副总裁William Fedus离职，将致力于AI for Science

William Fedus
@LiamFedus

This is what I sent to my colleagues at OpenAI:

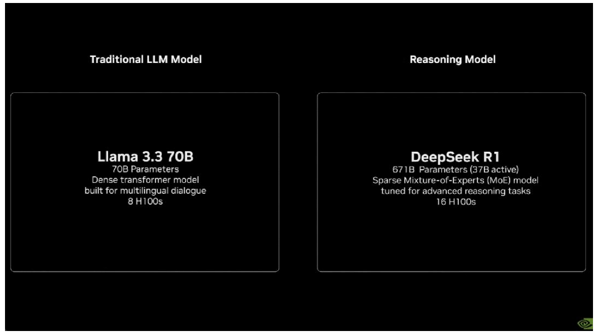
Hi all, I made the difficult decision to leave OpenAI as an employee, but I'm looking to work closely together as a partner going forward. Contributing to the mission of OpenAI and working with world-class teams to create and improve ChatGPT has been an experience of a lifetime.

But I've gotten really excited about AI for science. My undergrad was in physics and I'm keen to apply this technology there. Because AI for science is one of the most strategically important areas to OpenAI and achieving ASI, OpenAI is planning to invest in and partner with my new company. So I'll see you all around!

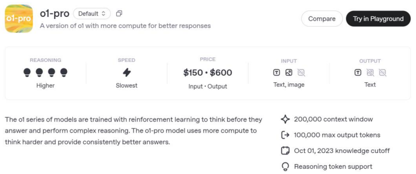
Thanks to all the leadership who believed in me early on, especially, Sam, Greg, and Mark. Thank you everyone on post-training and to all of our collaborators across research and product. I'll miss working with so many of you, but will be cheering you on! Post-training has an amazing roster of talent and leaders who will continue to drive its success.

4:59 AM · Mar 18, 2025 · 51K Views

Blackwell Ultra专为推理大模型加速



OpenAI推出o1-pro API，价格昂贵



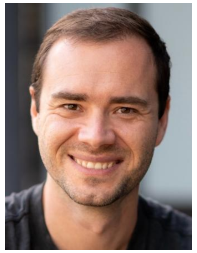
提示工程

CS2916 大语言模型

飲水思源 愛國榮校

https://plms.ai/teaching/index.html

提示工程



the hottest new programming language is English

Andrej Karpathy

未来的编程语言只会剩下两种：一种叫英文，一种叫中文

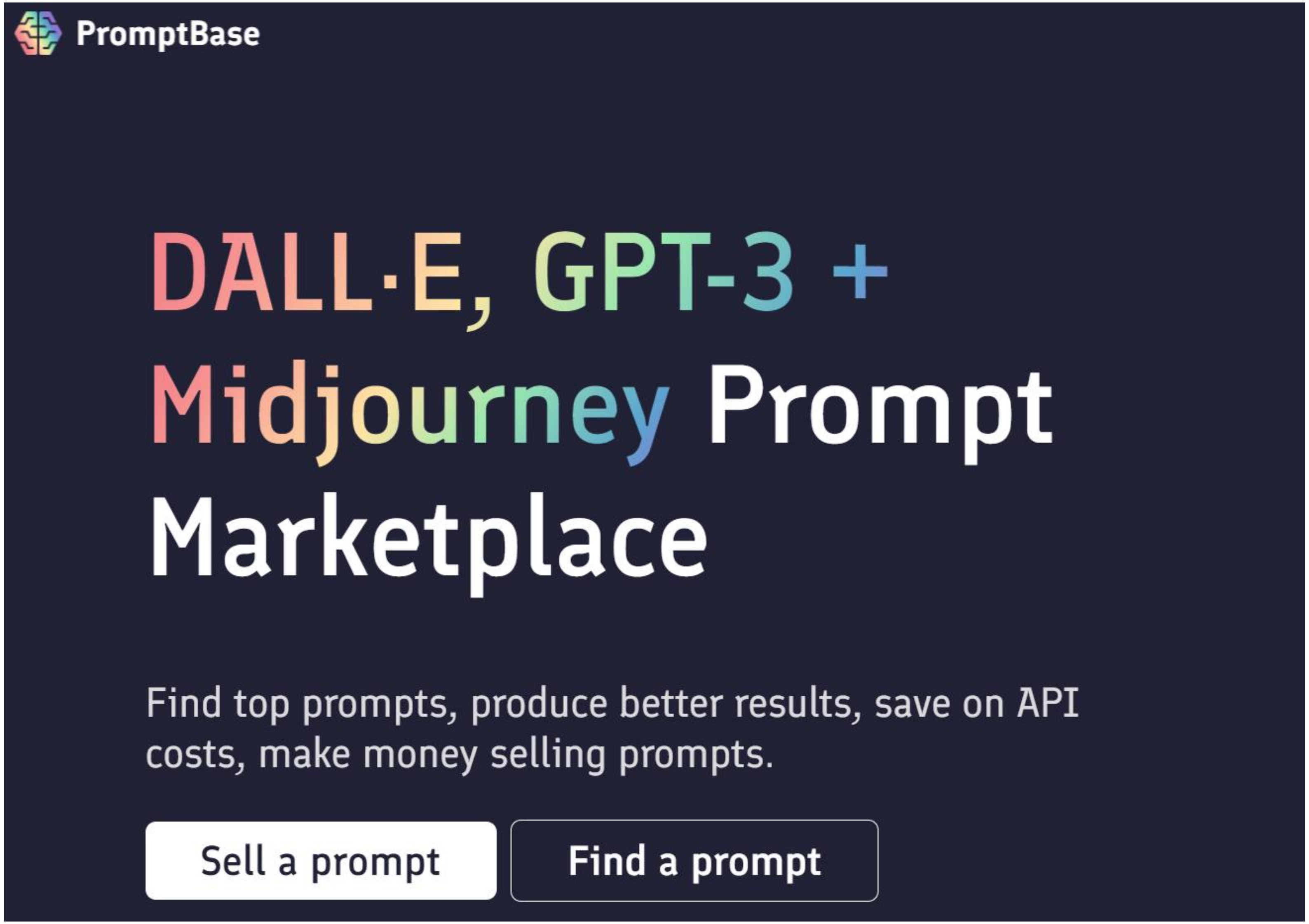
李彦宏

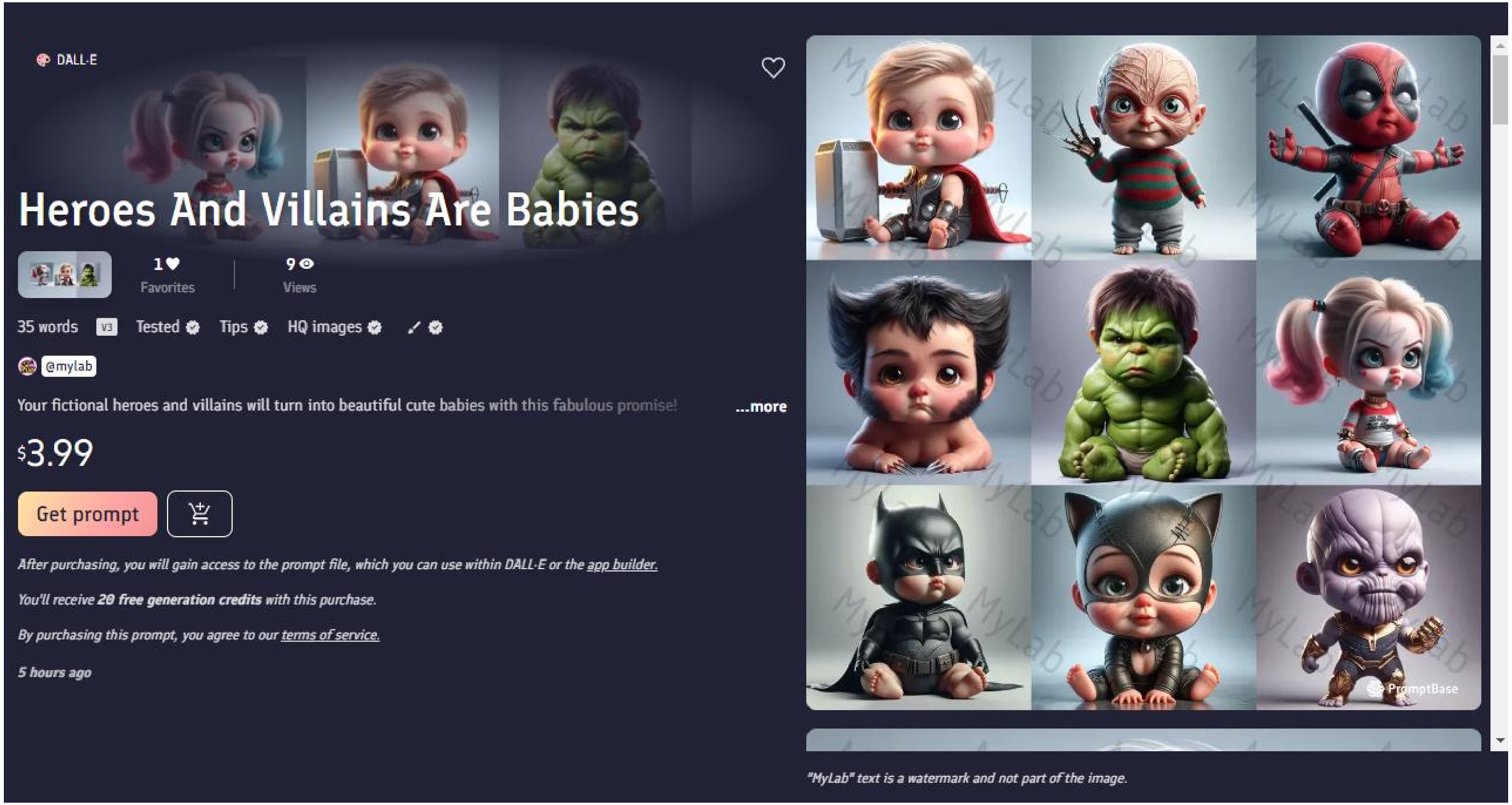
提示工程

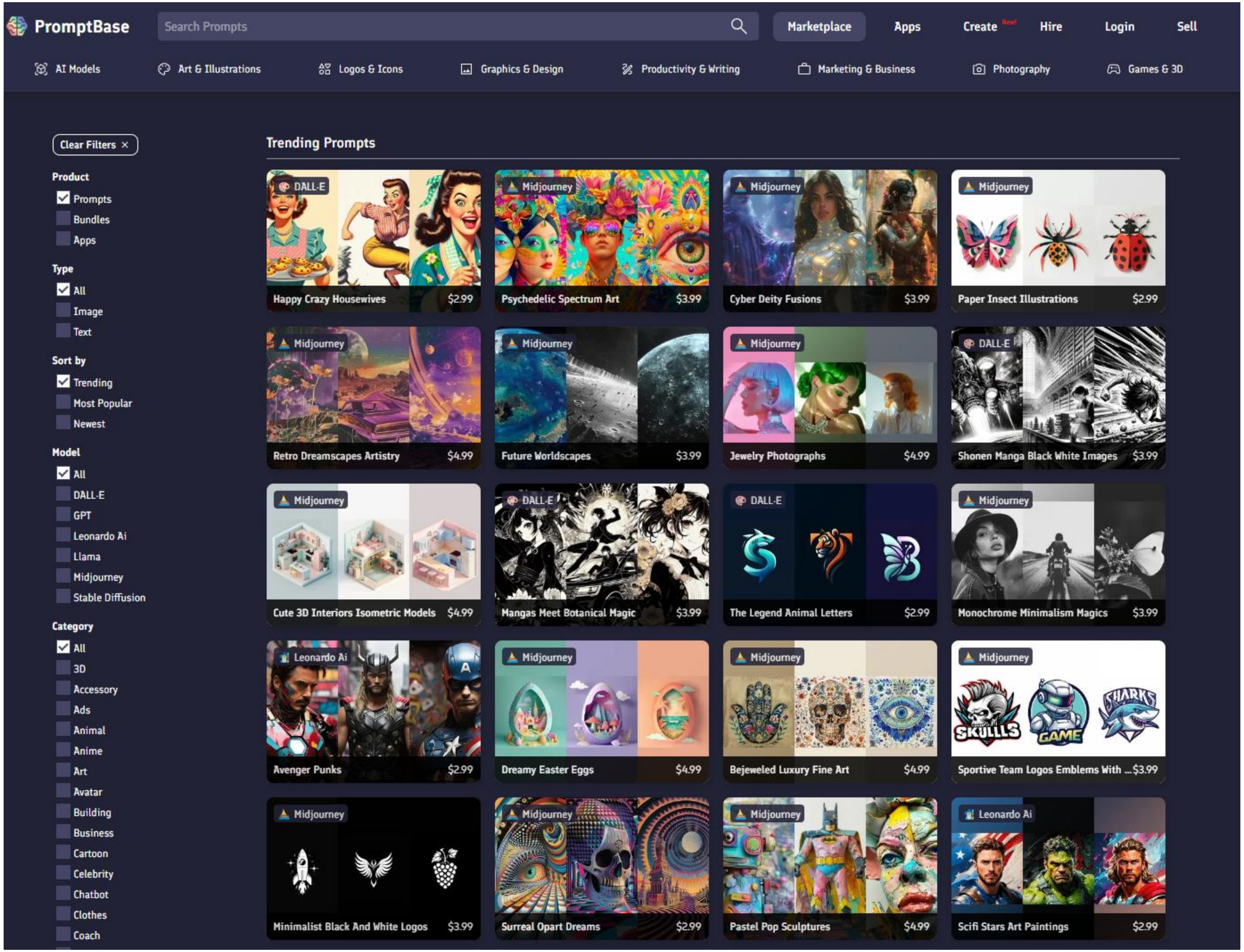
核心任务

- **任务简介**：给定一篇AI领域学术论文，考生需要设计并实现一个多层次任务分解系统，将论文转化为面向公众的科技新闻稿。新闻稿应当准确传达论文的核心发现和贡献，同时以生动、易懂的方式呈现，适合非专业读者阅读。
  - 考生将收到1篇最新发表的AI领域学术论文（PDF格式），需要通过设计一系列子任务和相应的提示策略，引导LLM逐步理解、提取和重构论文内容，最终生成一篇高质量的科技新闻稿。
  - 考生可以自由阅读、分析"一个好的新闻稿长什么样"（如机器之心、量子位等）
  - 考生可以自由选择使用任何厂家/版本的大模型或API，通过设计合理的任务分解流程和提示策略，完成从专业学术内容到公众科普内容的转化。
- **提供给考生的文件**：1篇AI领域学术论文的PDF文件（https://arxiv.org/pdf/2501.00747）
- **提交具体要求**：
  - 提交的附件包含两项内容：
    - 第一项：生成的新闻稿（markdown格式）

Adversarial Multi-task Learning for Text Classification P Liu, X Qiu, X Huang ACL 2017	785	2017
Extractive Summarization as Text Matching M Zhong, P Liu, Y Chen, D Wang, X Qiu, X Huang ACL 2020	576	2020


PromptBase
DALL·E, GPT-3 + Midjourney Prompt Marketplace
Find top prompts, produce better results, save on API costs, make money selling prompts.
Sell a prompt
Find a prompt


DALL·E
Heroes And Villains Are Babies
1 Favorites
9 Views
35 words
V3
Tested
Tips
HQ images
@mylab
Your fictional heroes and villains will turn into beautiful cute babies with this fabulous promise!
...more
$3.99
Get prompt
After purchasing, you will gain access to the prompt file, which you can use within DALL·E or the app builder.
You'll receive 20 free generation credits with this purchase.
By purchasing this prompt, you agree to our terms of service.
5 hours ago
PromptBase
"MyLab" text is a watermark and not part of the image.


PromptBase
Search Prompts
Marketplace
Apps
Create
Hire
Login
Sell
AI Models
Art & Illustrations
Logos & Icons
Graphics & Design
Productivity & Writing
Marketing & Business
Photography
Games & 3D
Clear Filters ×
Product
Prompts
Bundles
Apps
Type
All
Image
Text
Sort by
Trending
Most Popular
Newest
Model
All
DALL-E
GPT
Leonardo Ai
Llama
Midjourney
Stable Diffusion
Category
All
3D
Accessory
Ads
Animal
Anime
Art
Avatar
Building
Business
Cartoon
Celebrity
Chatbot
Clothes
Coach
Trending Prompts
DALL-E Happy Crazy Housewives $2.99
Midjourney Psychedelic Spectrum Art $3.99
Midjourney Cyber Deity Fusions $3.99
Midjourney Paper Insect Illustrations $2.99
Midjourney Retro Dreamscapes Artistry $4.99
Midjourney Future Worldscapes $3.99
Midjourney Jewelry Photographs $4.99
DALL-E Shonen Manga Black White Images $3.99
Midjourney Cute 3D Interiors Isometric Models $4.99
DALL-E Mangas Meet Botanical Magic $3.99
DALL-E The Legend Animal Letters $2.99
Midjourney Monochrome Minimalism Magics $3.99
Leonardo Ai Avenger Punks $2.99
Midjourney Dreamy Easter Eggs $4.99
Midjourney Bejeweled Luxury Fine Art $4.99
Midjourney Sportive Team Logos Emblems With ... $3.99
Midjourney Minimalist Black And White Logos $3.99
Midjourney Surreal Opart Dreams $2.99
Midjourney Pastel Pop Sculptures $4.99
Leonardo Ai Scifi Stars Art Paintings $2.99

What is the “Prompt”?

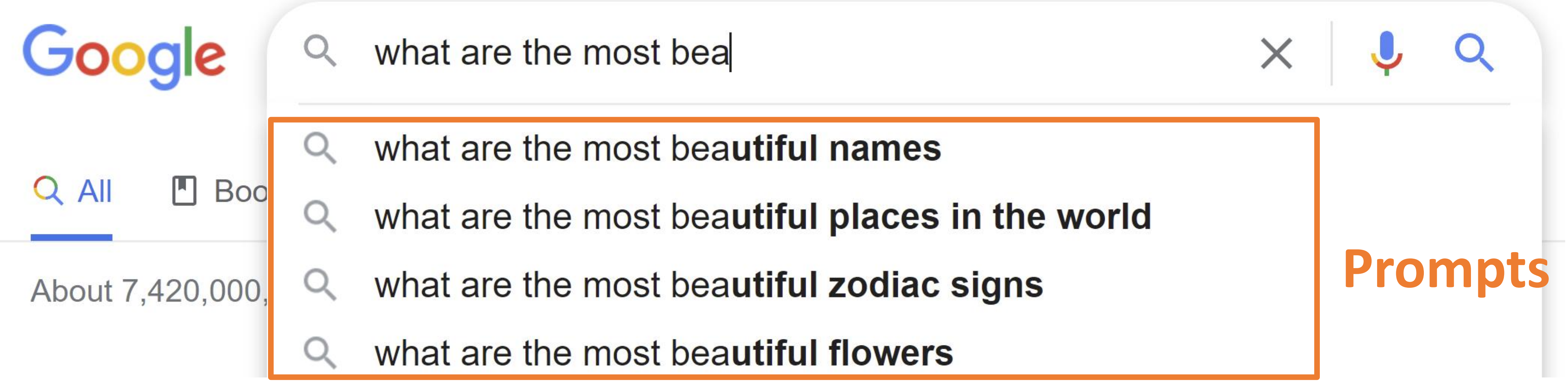
Prompt meaning prŏmpt

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

Secret in Modern NLP Development

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
Task Models

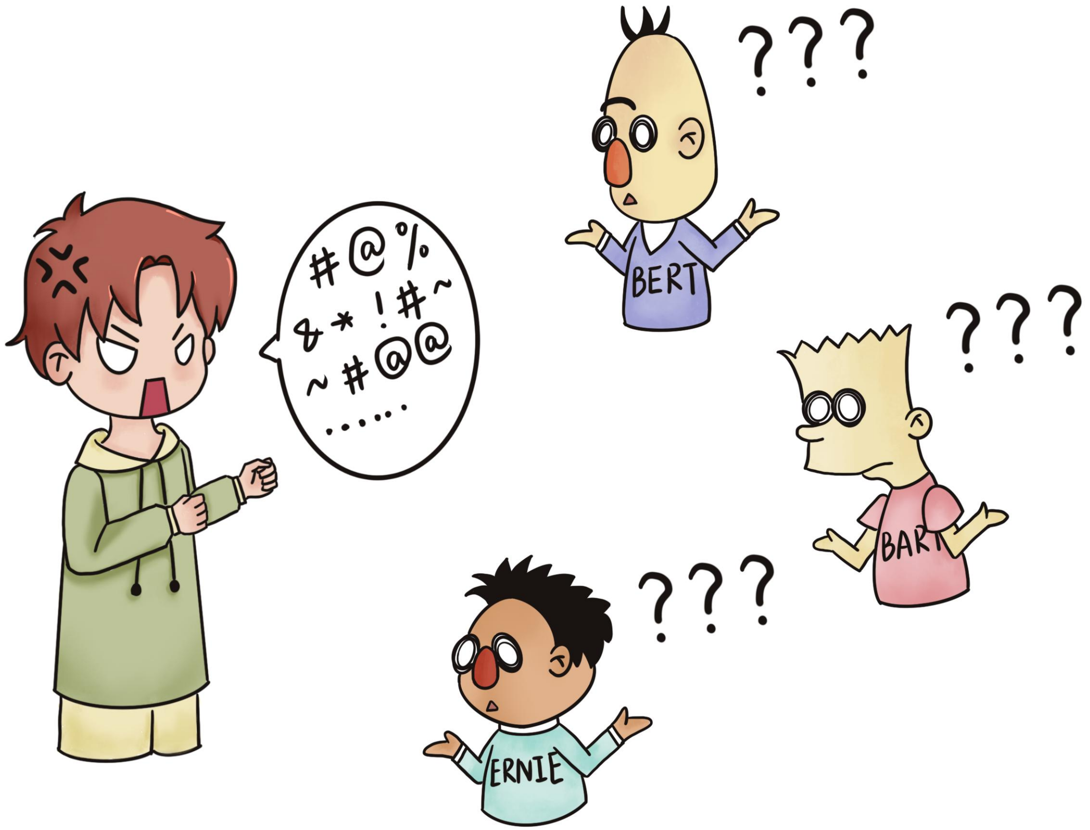
Pre-trained
Language Models

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

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

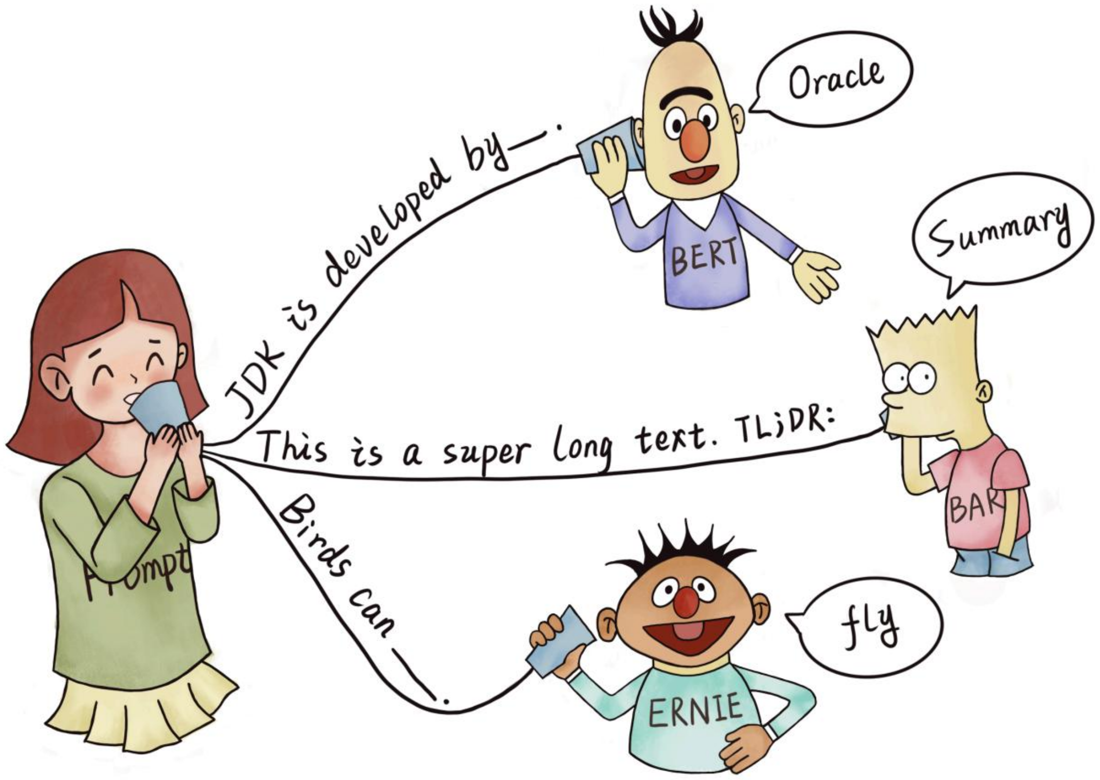
直观的定义

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



直观的定义

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



更技术层面的定义

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

purpose

Method

更技术层面的定义

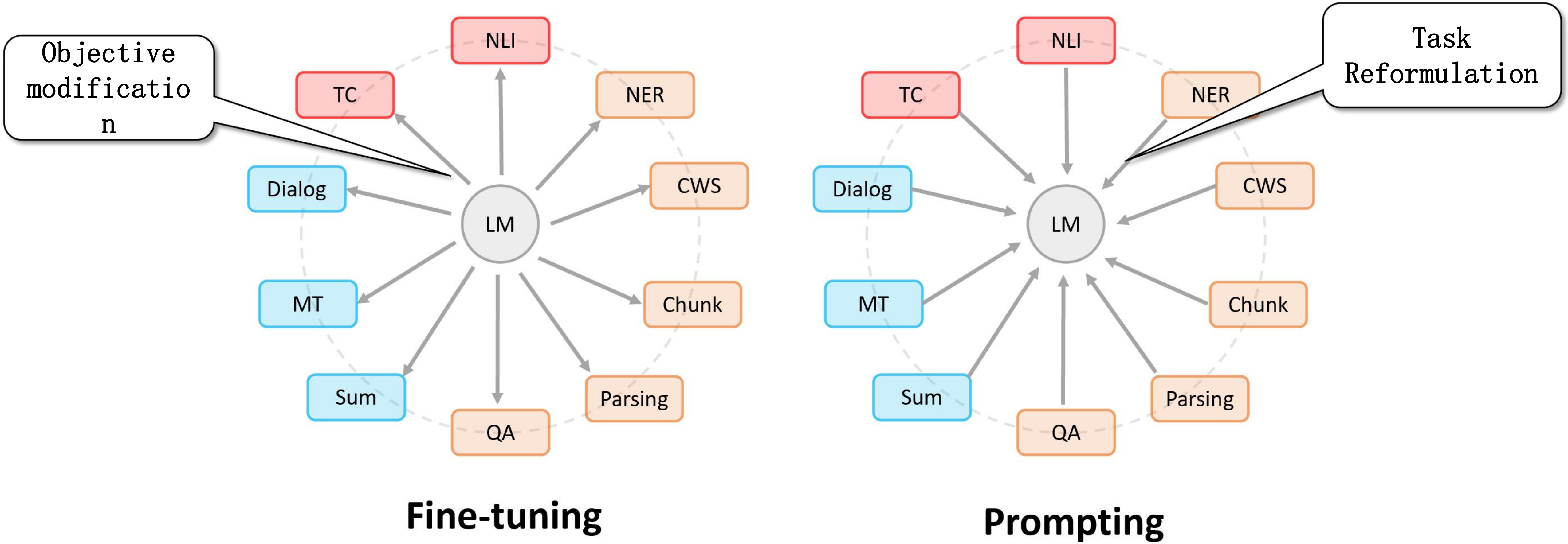
- Prompt is the technique of making better use of the knowledge from the pre-trained model by adding additional texts to the input.

purpose

Method

还有什么好处?

任务的“大一统”



Objective
modificatio
n
NLI
TC
NER
Dialog
CWS
LM
MT
Chunk
Sum
Parsing
QA
Fine-tuning
Task
Reformulation
NLI
TC
NER
Dialog
CWS
LM
MT
Chunk
Sum
Parsing
QA
Prompting

What is the general workflow of prompt-based methods?

Prompting for Sentiment Classification

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

Input: x = I love this movie.

Prompting for Sentiment Classification

- 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.

Prompting for Sentiment Classification

- 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.

Prompting for Sentiment Classification

- 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.

Prompting for Sentiment Classification

- 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.

Prompting for Sentiment Classification

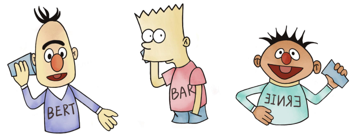
- Given a prompt, predict the answer [z].
  - Choose a suitable pretrained language model;

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?

Prompting for Sentiment Classification

- 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.

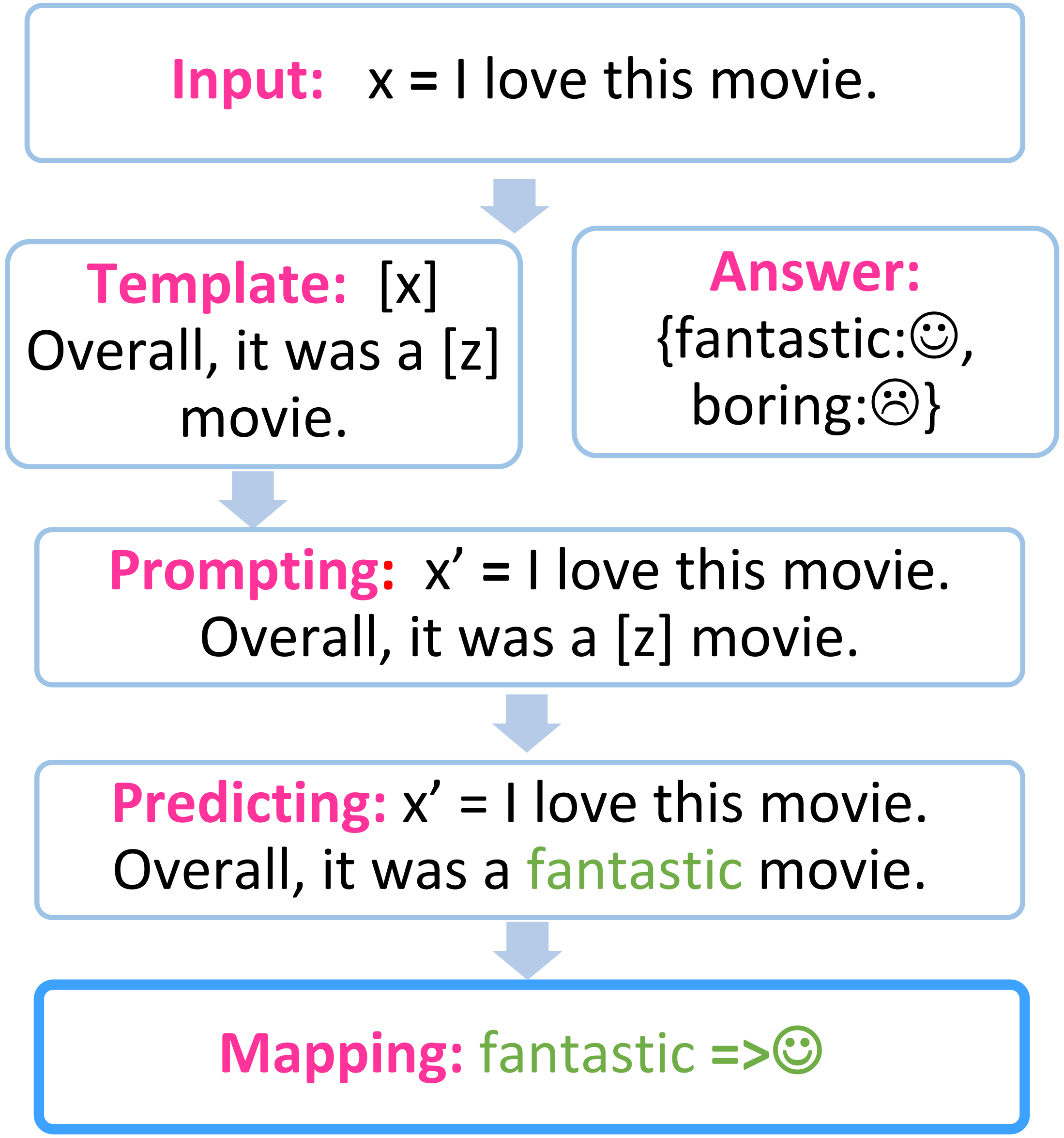
Answer: {fantastic:☺, boring:☹}

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

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

Prompting for Sentiment Classification

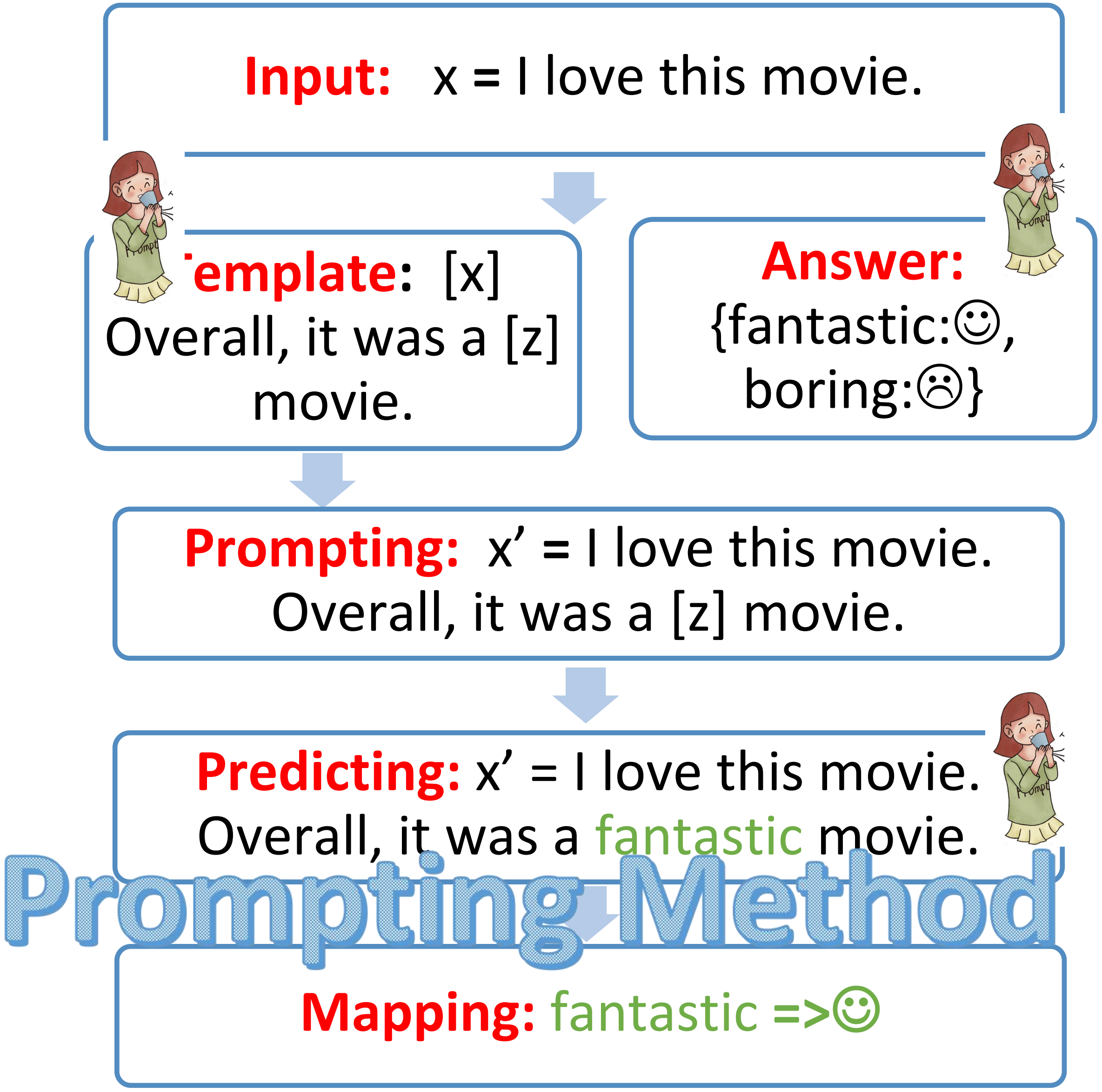
- Mapping: Given an answer, map it into a class label.
  - fantastic => ☺



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 =>☺

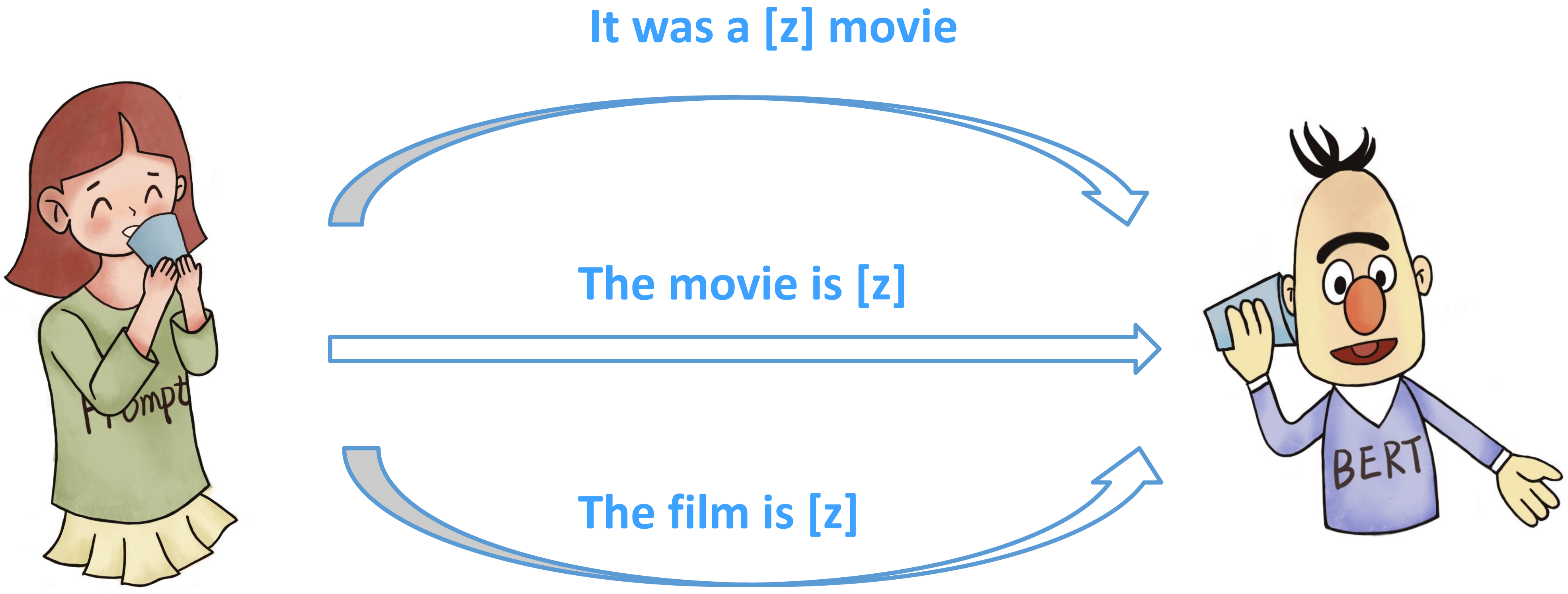
What are the design considerations for prompt-based methods?

Design Considerations for Prompt-based Methods

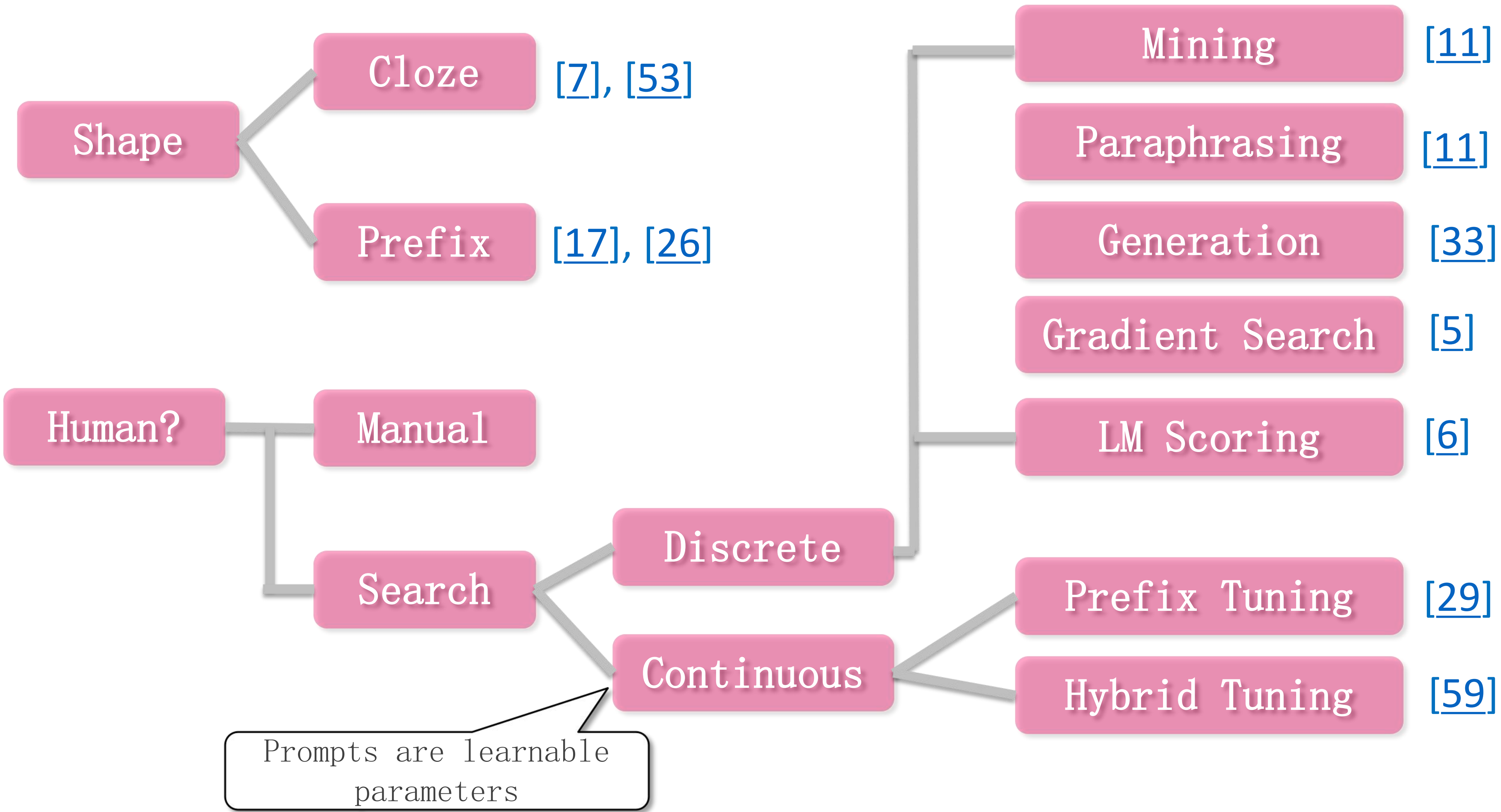
- Prompt Template Engineering
- Answer Engineering
- Pre-trained Model Choice
- Expanding the Paradigm
- Prompt-based Training Strategies

Prompt Template Engineering

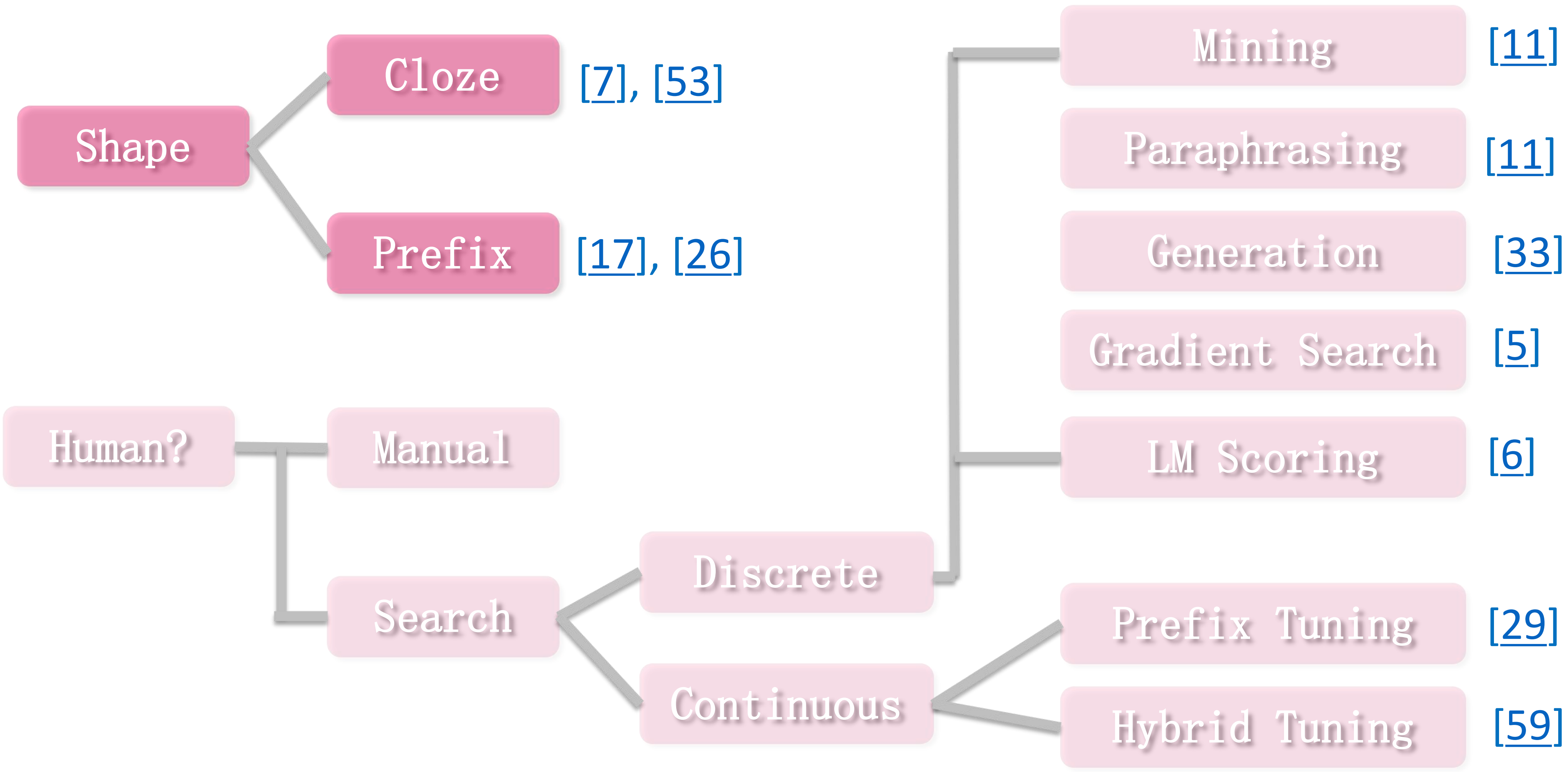
- Research Question:
  - how to define appropriate prompt templates



Design Decision of Prompt Templates



Design Decision of Prompt Templates



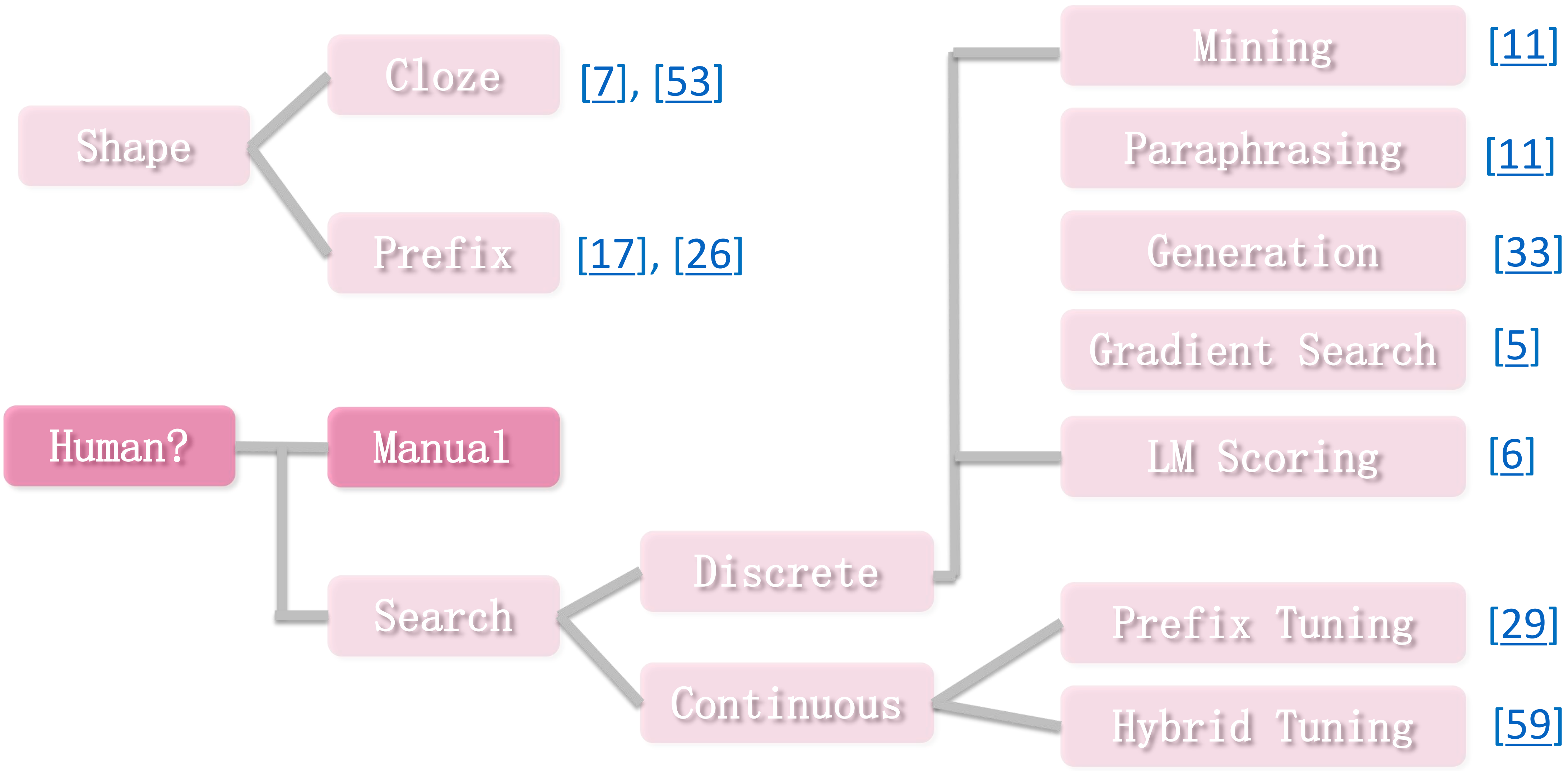
Prompt Shape

- Cloze Template
  - Contain blanks to be filled.
  - Useful for Masked LMs.
    - *" The capital of ____ is Beijing ."*

Prompt Shape

- Cloze Template
- Prefix Template
  - Contain a string prefix to be continued.
  - Useful for Left-to-right LM and Encoder-Decoder LM.
    - *"President Joe Biden and three of his European allies face …… TL;DR: ____ "*

Prompt Shape



Manual Template Design

- Manual Prompt
  - The most natural way to create prompts
    - I love this movie so much! What’s the sentiment of the text? ___ .
    - President Joe Biden and three of his European allies face …… In summary, ___ .
    - President Joe Biden and three of his European allies face …… The article is about ___ .

Manual Template Design

Manual Prompt

- The most natural way to create prompts
- An art that takes time and experience.

Zero-shot Accuracy (BERT-base, SST-2)

- First template–answer pair

Template: <A movie review> The movie is ___ .

Answer: fantastic/terrible

0.749

- Second template–answer pair

Template: <A movie review> The review is ___ .

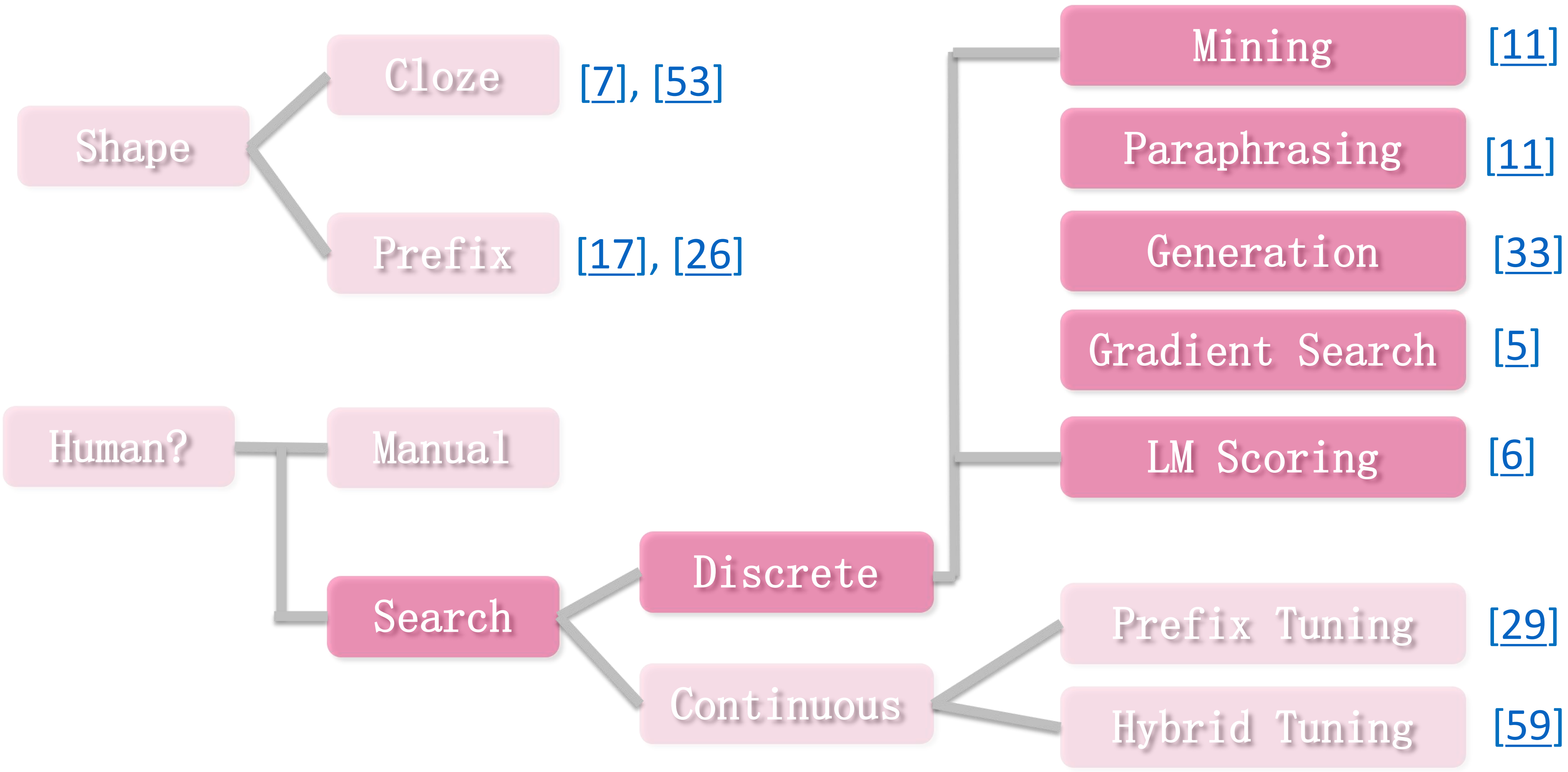
Answer: positive/negative

0.534

Manual Template Design

- Manual Prompt
  - The most natural way to create prompts
  - An art that takes time and experience.
  - For some complicated tasks, its hard to manually craft templates.

Design Decision of Prompt Templates



Discrete Search

- Mining
- Paraphrasing
- Gradient-based Search
- Generation
- LM Scoring

Discrete Search

- Mining
  - Use a large corpus to mine templates that contain both the input and the gold answer.
  - Example
    - Fact retrieval for country-capital relationship
    - search through Wikipedia and find strings that contain both ``Beijing” and ``China” or other pairs.

Input	Gold answer
China	Beijing
Japan	Tokyo
United States	Washington

- Beijing, the capital of China
- The capital of China is Beijing
- ……

Reference: Jiang et al. How Can We Know What Language Models Know? TACL(2020)

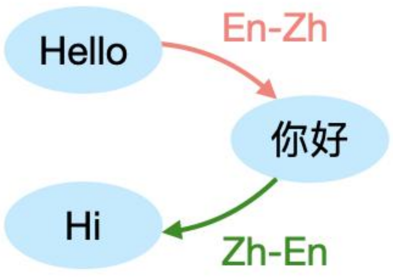
Discrete Search

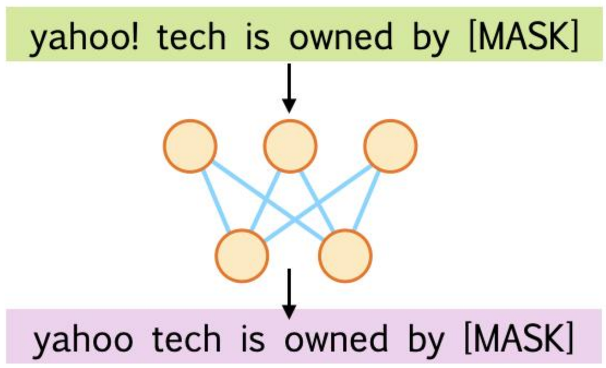
- Paraphrasing
  - Take in an existing seed template, and paraphrases it into a set of other candidate templates.

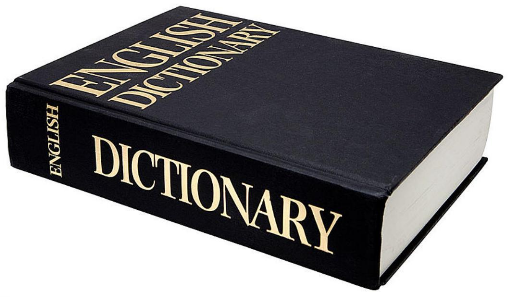
References: [1] Jiang et al. How Can We Know What Language Models Know? TACL (2020). [2] Yuan et al. BARTScore: Evaluating Generated Text as Text Generation. NeurIPS (2021). [3] Haviv et al. BERTese: Learning to Speak to BERT. EACL (2021).

Discrete Search

- Paraphrasing
  - Take in an existing seed template, and paraphrases it into a set of other candidate templates.
  - Typical methods
    - Back-translation
    - Using replacement of phrases from a thesaurus
    - Use neural rewriter to rewrite







References: [1] Jiang et al. How Can We Know What Language Models Know? TACL (2020). [2] Yuan et al. BARTScore: Evaluating Generated Text as Text Generation. NeurIPS (2021). [3] Haviv et al. BERTese: Learning to Speak to BERT. EACL (2021).

Discrete Search

- Gradient-based Search
  - Stepping through tokens and find ones that can trigger desired outputs.

I love this movie! $\boxed{}$ ___ .

We want the LM to predict **positive** here

The **template token** we want to search

Discrete Search

- Gradient-based Search
  - Stepping through tokens and find ones that can trigger desired outputs.

I love this movie! $\boxed{}$ ____ .

We want the LM to predict **positive** here

Token	P(positive)
is	0.8
hello	0.09
cat	0.04
….	…

Reference: Shin et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP (2020).

Discrete Search

Gradient-based Search

- Stepping through tokens and find ones that can trigger desired outputs.

I love this movie! ☐ ___ . ← We want the LM to predict **positive** here

Token	P(positive)
is	0.8
hello	0.09
cat	0.04
….	…

Reference: Shin et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP (2020).

Discrete Search

- Generation
  - Use LM to generate templates.

Pre-train

Input: Thank you <X> me to the party <Y> week.

Target: <X> for inviting <Y> last <Z>

Reference: Shin et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP (2020).

Discrete Search

- Generation
  - Use LM to generate templates.

I love this movie! <X> great <Y>

↓ T5 decode

<X> This is <Y> . <Z>

<X> A <Y> one. <Z>

……

Reference: Shin et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP (2020).

Discrete Search

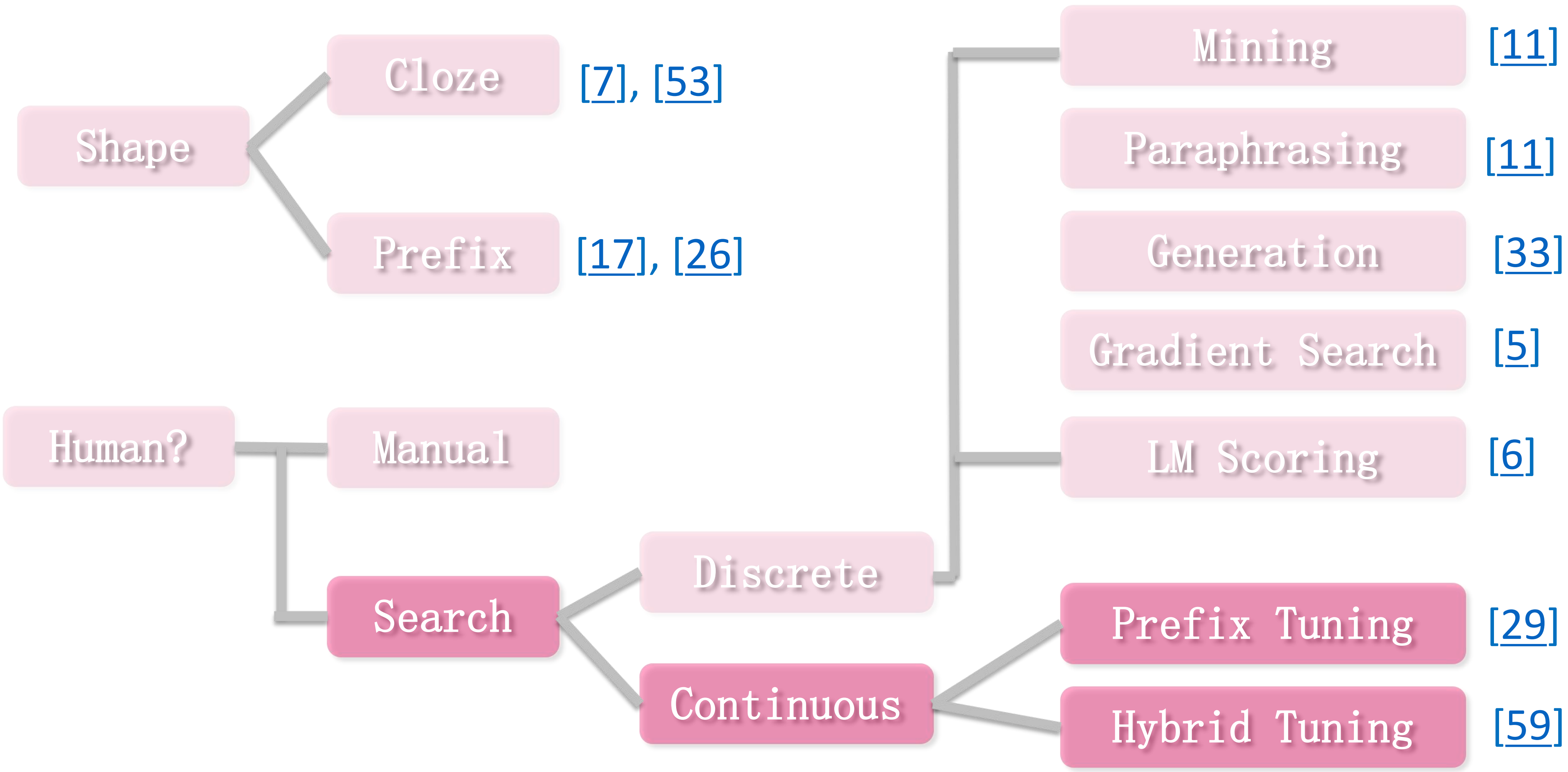
- LM Scoring
  - Use the LM to choose the templates that achieve high LM probability.

I love this movie! <template> positive.

Sequence	P
I love this movie! The sentiment of the text is positive.	0.4
I love this movie! Hello world positive	0.09
I love this movie! The text is positive	0.3
….	…

Reference: Davison et al. Commonsense Knowledge Mining from Pretrained Models. EMNLP (2019).

Design Decision of Prompt Templates



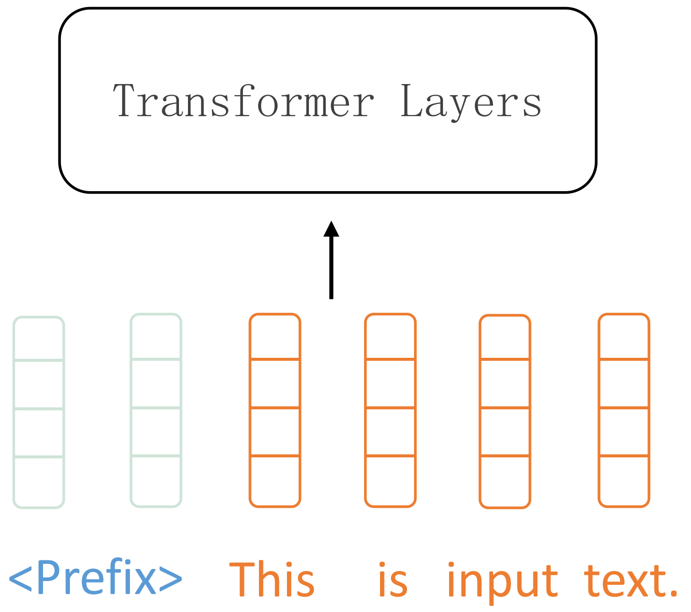
Continuous Template Search

- Prefix Tuning
  - Prepends a sequence of continuous task-specific vectors to the input, while keeping the LM parameters frozen.

References: [1] Li et al. Prefix-Tuning: Optimizing Continuous Prompts for Generation. arXiv:2101.00190 (2021). [2] Lester et al. The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691 (2021)

Continuous Template Search

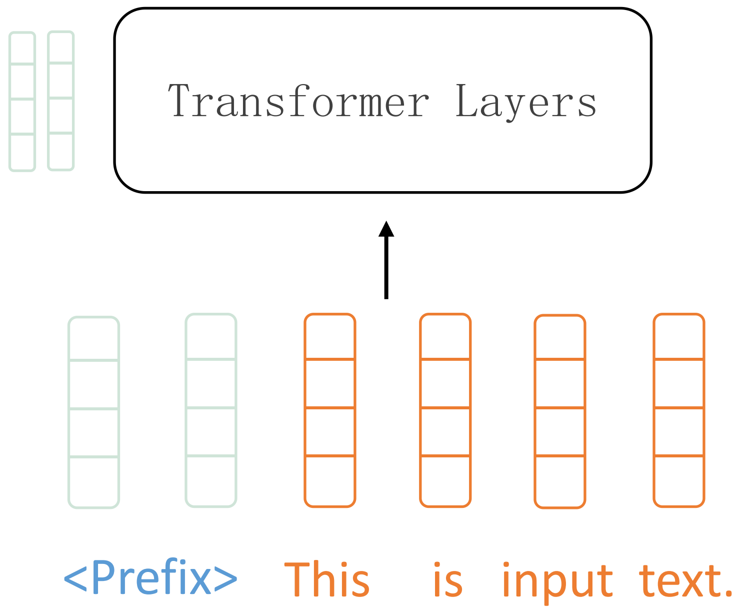
- Prefix Tuning
  - Prepends a sequence of continuous task-specific vectors to the input, while keeping the LM parameters frozen.
    - Shallow Prefix Tuning



References: [1] Li et al. Prefix-Tuning: Optimizing Continuous Prompts for Generation. arXiv:2101.00190 (2021). [2] Lester et al. The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691 (2021)

Continuous Template Search

- Prefix Tuning
  - Prepends a sequence of continuous task-specific vectors to the input, while keeping the LM parameters frozen.
    - Shallow Prefix Tuning
    - Deep Prefix Tuning



References: [1] Li et al. Prefix-Tuning: Optimizing Continuous Prompts for Generation. arXiv:2101.00190 (2021). [2] Lester et al. The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691 (2021)

Continuous Template Search

- Hybrid Tuning
  - An extension of prefix tuning

References: [1] Zhong et al. Factual Probing Is [MASK]: Learning vs. Learning to Recall. CoRR abs/2104.05240 (2021). [2] Liu et al. GPT Understands, Too. CoRR abs/2103.10385 (2021).

Continuous Template Search

- Hybrid Tuning
  - An extension of prefix tuning
  - The positions of tunable virtual tokens can be anywhere.

I love this movie so much! positive.

References: [1] Zhong et al. Factual Probing Is [MASK]: Learning vs. Learning to Recall. CoRR abs/2104.05240 (2021). [2] Liu et al. GPT Understands, Too. CoRR abs/2103.10385 (2021).

Continuous Template Search

- Hybrid Tuning
  - An extension of prefix tuning
  - The positions of tunable virtual tokens can be anywhere.
  - Use hard templates initialization

I love this movie so much! The sentiment is positive.

References: [1] Zhong et al. Factual Probing Is [MASK]: Learning vs. Learning to Recall. CoRR abs/2104.05240 (2021). [2] Liu et al. GPT Understands, Too. CoRR abs/2103.10385 (2021).

Continuous Template Search

- Hybrid Tuning
  - An extension of prefix tuning
  - The positions of tunable virtual tokens can be anywhere.
  - Use hard templates initialization
  - Combine hard and soft template tokens

I love this movie so much! is positive.

References: [1] Zhong et al. Factual Probing Is [MASK]: Learning vs. Learning to Recall. CoRR abs/2104.05240 (2021). [2] Liu et al. GPT Understands, Too. CoRR abs/2103.10385 (2021).

Design Considerations for Prompt-based Methods

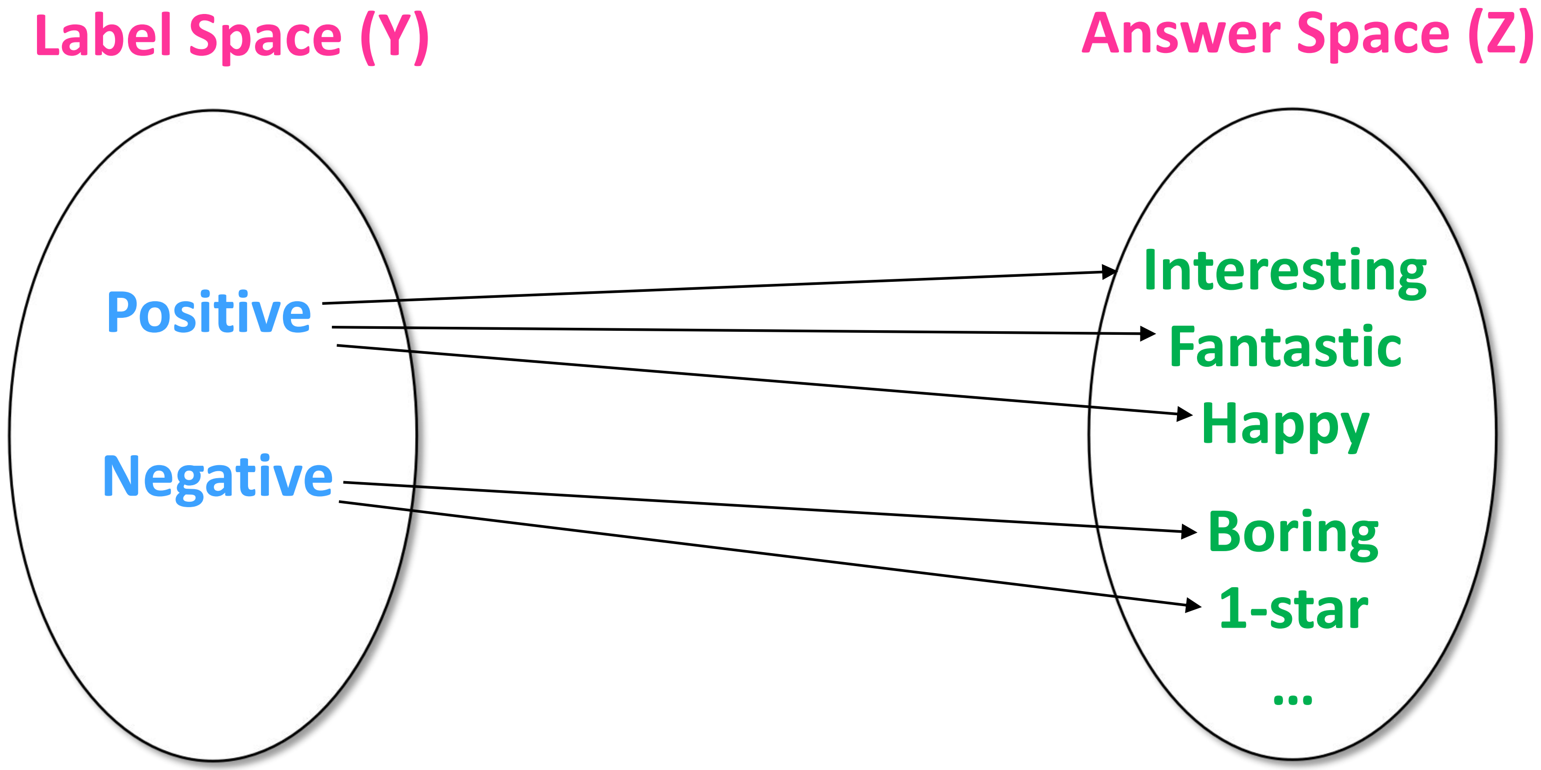
- Prompt Template Engineering
- Answer Engineering
- Pre-trained Model Choice
- Expanding the Paradigm
- Prompt-based Training Strategies

Answer Engineering

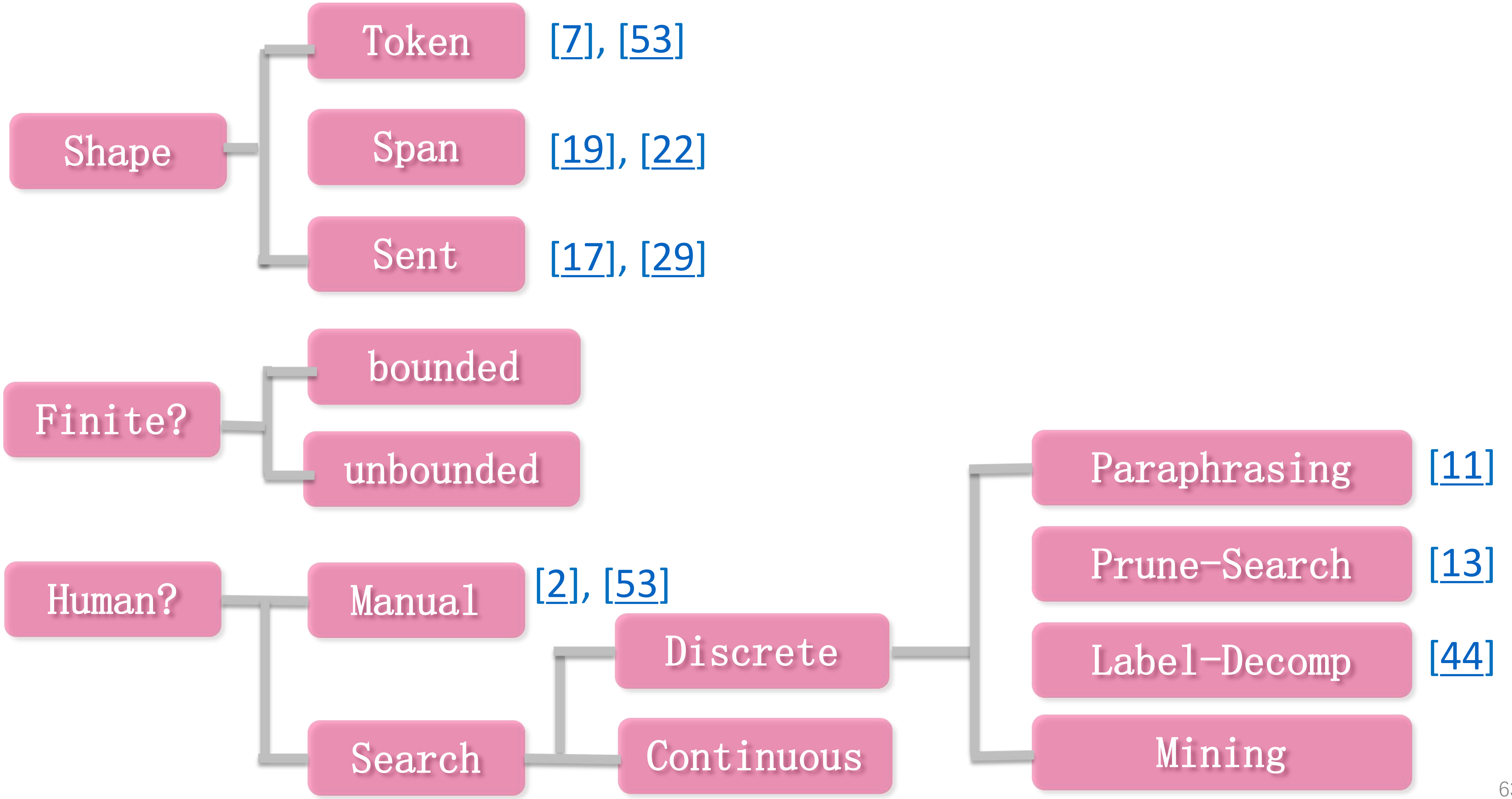
- Research Question:
  - Given a task (or a prompt), how to define a suitable mapping function between label space and answer space?

Answer Engineering

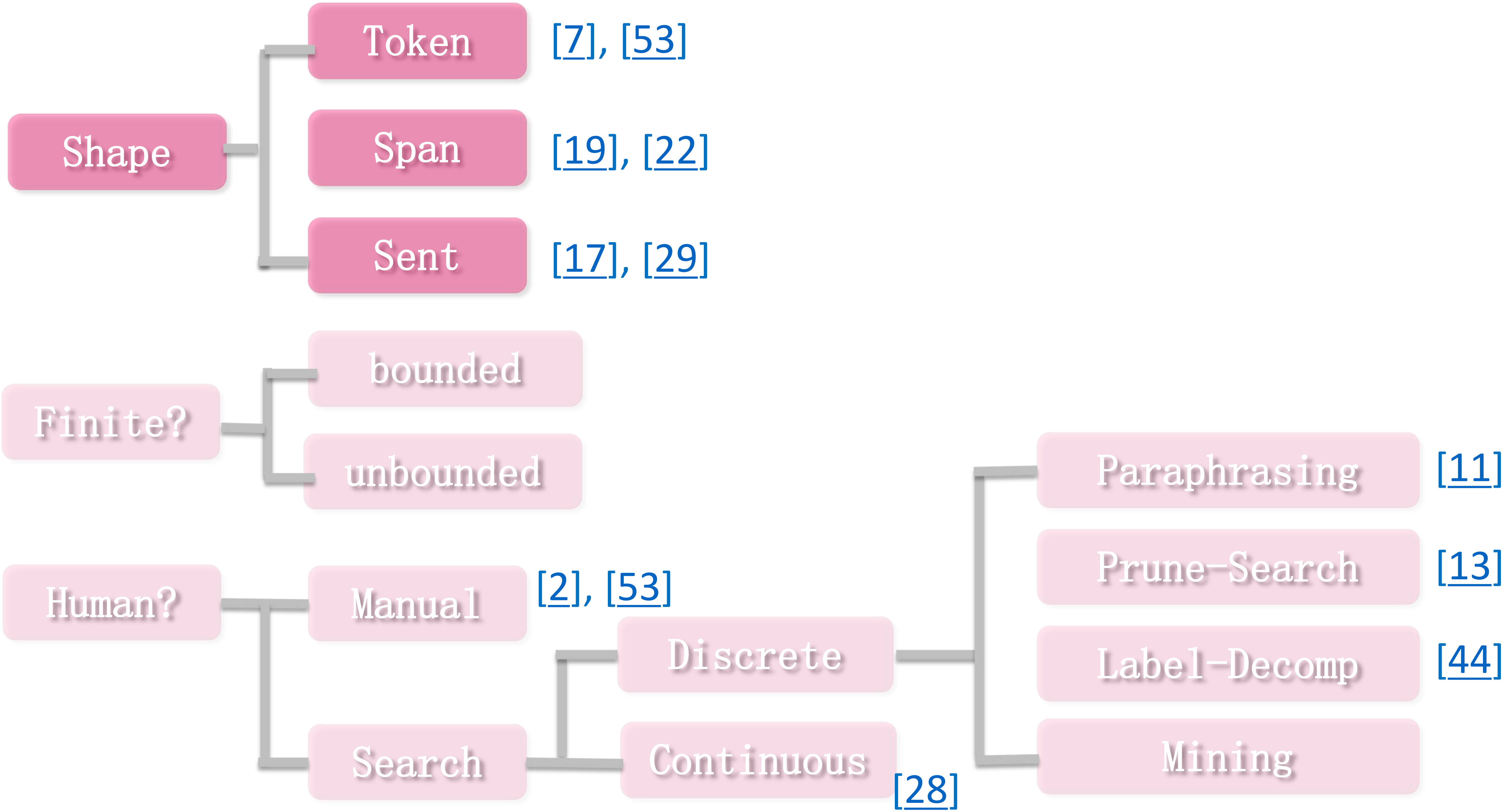
- Research Question:
  - Given a task (or a prompt), how to define a suitable mapping function between label space and answer space?



Design Decision of Prompt Answer Engineering



Design Decision of Prompt Answer Engineering



Design Considerations for Prompt-based Methods

- Token
  - Useful for most classification tasks
  - Examples
    - <A movie review> The movie is fantastic/terrible.
    - <Premise> Yes/No. <Hypothesis>

Design Considerations for Prompt-based Methods

- Token
- Span
  - Useful for classification with long label names, QA, knowledge probing, etc.
  - Example
    - Multiple choice QA

      A student riding a bicycle observes that it moves faster on a smooth road than on a rough road. This happens because the smooth road has

      (A) less gravity

      (B) more gravity

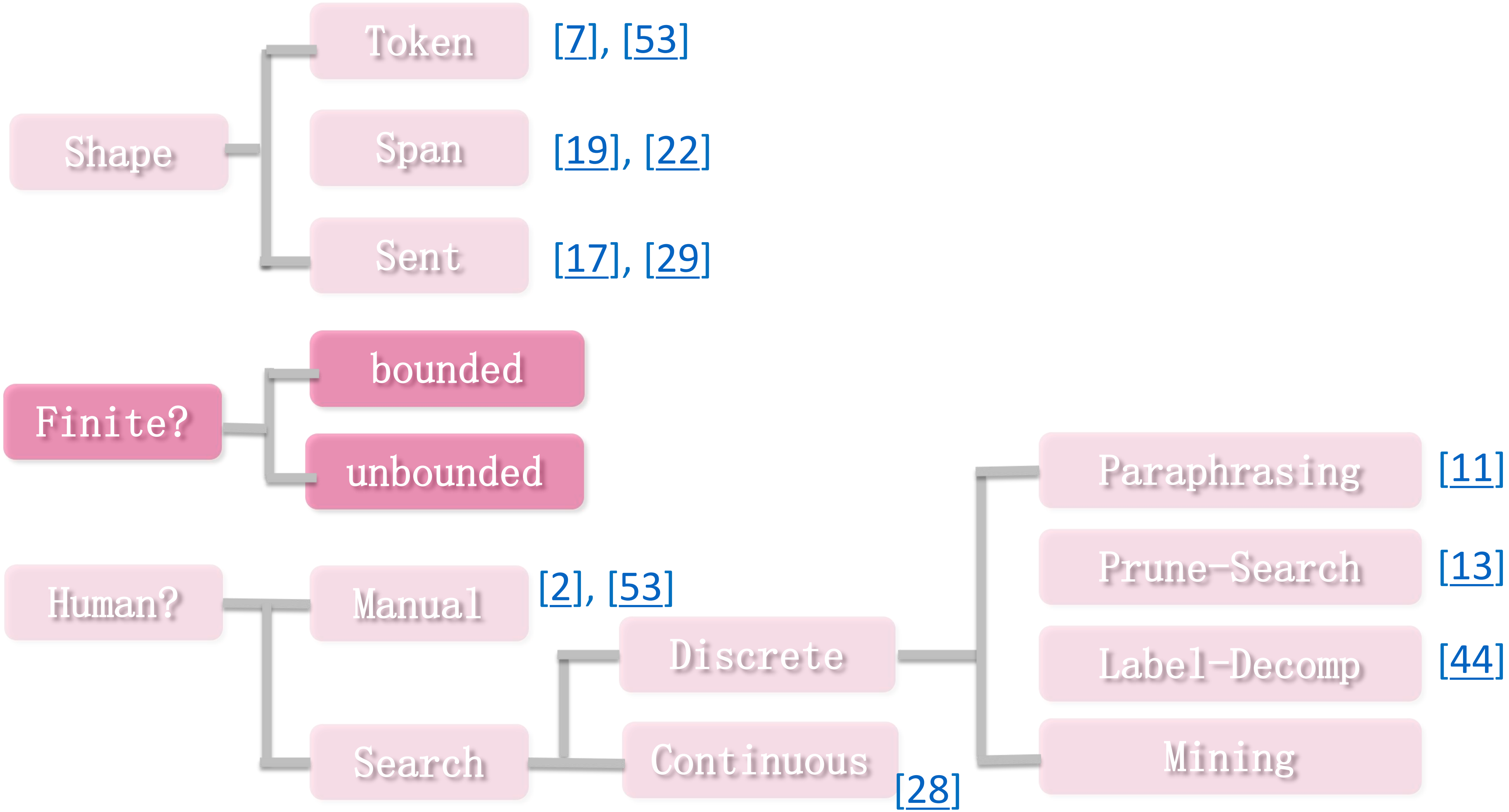
      (C) less friction [gold]

      (D) more friction

Design Considerations for Prompt-based Methods

- Token
- Span
- Sentence(s)
  - Useful for generation tasks, like MT or summarization.
  - Example
    - Translation from English to Chinese
      Input: Hello, world!
      Target (gold answer): 你好，世界!

Design Decision of Prompt Answer Engineering



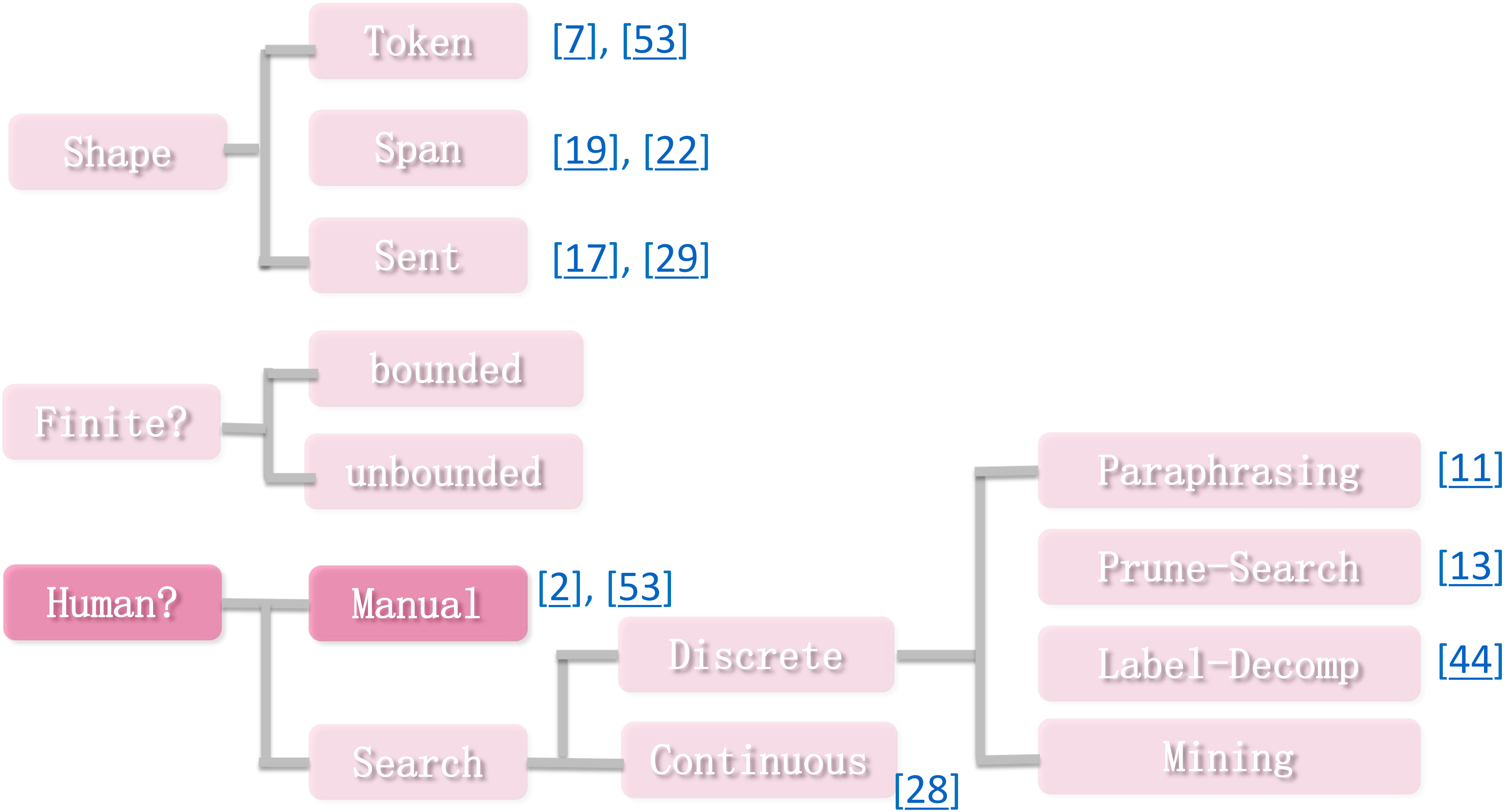
Answer Space

- Bounded
  - The space of possible outputs is constrained/finite.
  - Example
    - Text classification: health; finance; politics, sports.

Answer Space

- Bounded
  - The space of possible outputs is constrained/finite.
  - Example
    - Text classification: health; finance; politics, sports.
- Unbounded
  - The space of possible outputs is unconstrained/infinite.
  - Example
    - Text summarization: all valid sequence of tokens.

Design Decision of Prompt Answer Engineering



Human Design

- The most natural way to create answers
  - For generation tasks, we can use identity mapping to map target output directly to gold answer
    - In MT/Summarization, take the target directly as gold answer

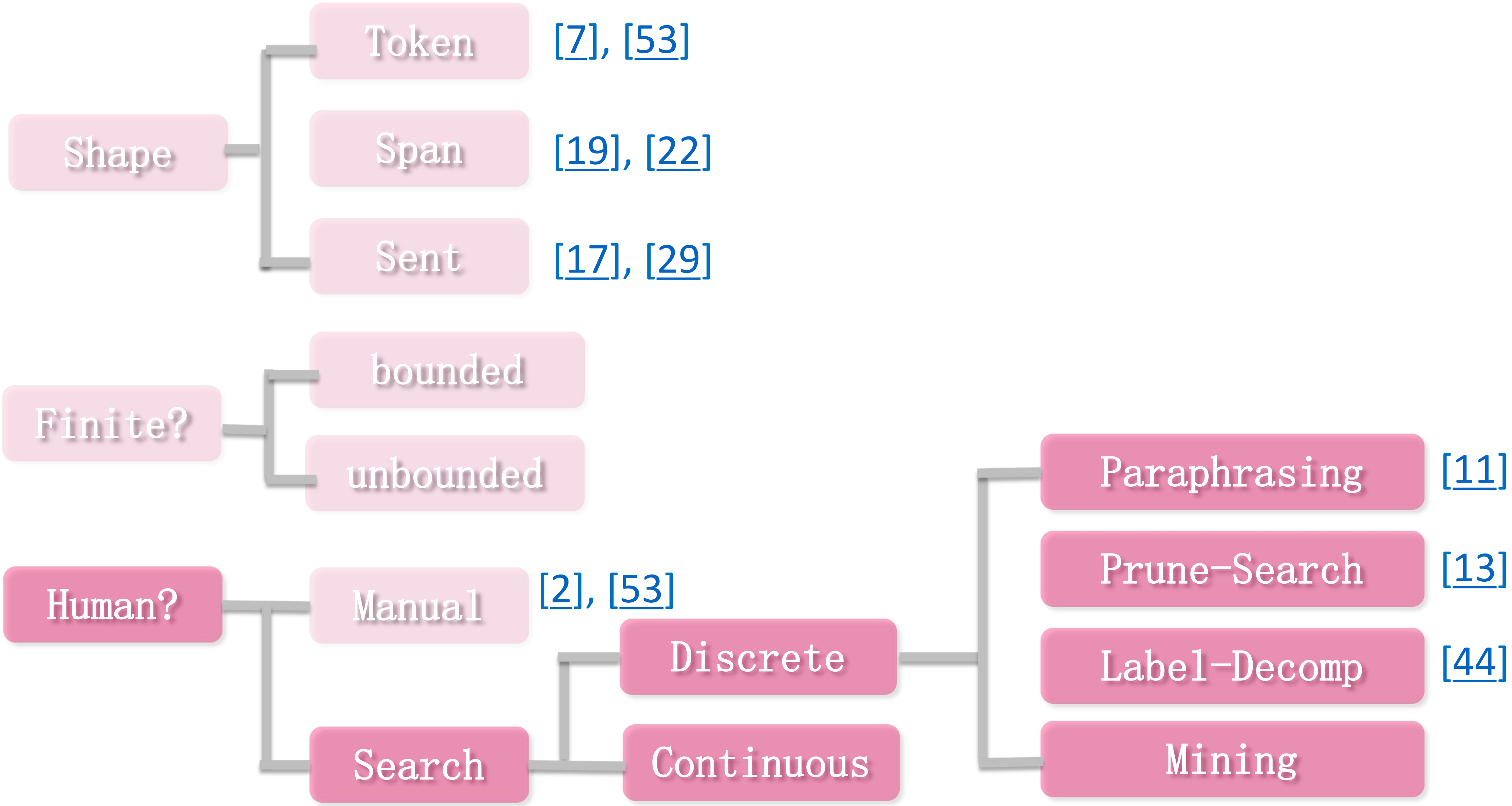
Human Design

- The most natural way to create answers
  - For generation tasks, we can use identity mapping to map target output directly to gold answer
    - In MT/Summarization, take the target directly as gold answer
  - For classification tasks, the label name can also act as gold answer.
    - For example, sports, politics

Human Design

- The most natural way to create answers
  - For generation tasks, we can use identity mapping to map target output directly to gold answer
    - In MT/Summarization, take the target directly as gold answer
  - For classification tasks, the label name can also act as gold answer.
    - For example, sports, politics
- An art that takes time and experience.
  - For some complicated tasks, it’s hard to manually craft answers.
    - For example, relation classification

Design Decision of Prompt Answer Engineering



Discrete Answer Search

- Paraphrasing
- Prune then Search
- Label Decomposition
- Mining

Discrete Answer Search

- Paraphrasing
  - Start with an initial answer space, and then use paraphrasing to expand this answer space to broaden its coverage.
  - Example
    - Multiple Choice QA

      A person wants to submerge himself in water, what should he use?

      (A) Whirl pool (Paraphrase to get Bathtub, A bathtub etc.)

      (B) …

Reference: Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2020. How Can We Know When Language Models Know? CoRR abs/2012.00955 (2020).

Discrete Answer Search

- Prune then Search
  - Pruning methods:
    - Select the most frequent words
    - Select tokens that have highest generation probability at answer position

References:
[1] Taylor Shin, Yasaman Razeghi, Robert L. LoganIV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Empirical Methods in Natural Language Processing (EMNLP).
[2] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In Association for Computational Linguistics (ACL).

Discrete Answer Search

- Prune then Search
  - Pruning methods:
    - Select the most frequent words
    - Select tokens that have highest generation probability at answer position
  - Searching methods:
    - Choose answers that maximize the likelihood of training data
    - Choose answers that achieve the best zero-shot accuracy

References:
[1] Taylor Shin, Yasaman Razeghi, Robert L. LoganIV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Empirical Methods in Natural Language Processing (EMNLP).
[2] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In Association for Computational Linguistics (ACL).

Discrete Answer Search

Label Decomposition

- For complex label, decompose the label into its constituent words.
- Example

  - Text classification:

    Science and Mathematics $\xrightarrow{\text{decompose}}$ {Science, Mathematics}

  - Relation Extraction:

    city_of_death $\xrightarrow{\text{decompose}}$ {person, city, death}

Reference: Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. AdaPrompt: Adaptive Prompt-based Finetuning for Relation Extraction. CoRR abs/2104.07650 (2021) .

Discrete Answer Search

- Mining
  - Given a seed answer, use some knowledge base to retrieve related words.
  - Example: “city”
    - metropolis town
    - urban
    - suburb
    - municipal
    - downtown
    - Country
    - ……

Reference: Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li and Hong-Gee Kim. 2021. Prompt-Learning for Fine-Grained Entity Typing. CoRR abs/2108.10604 (2021) .

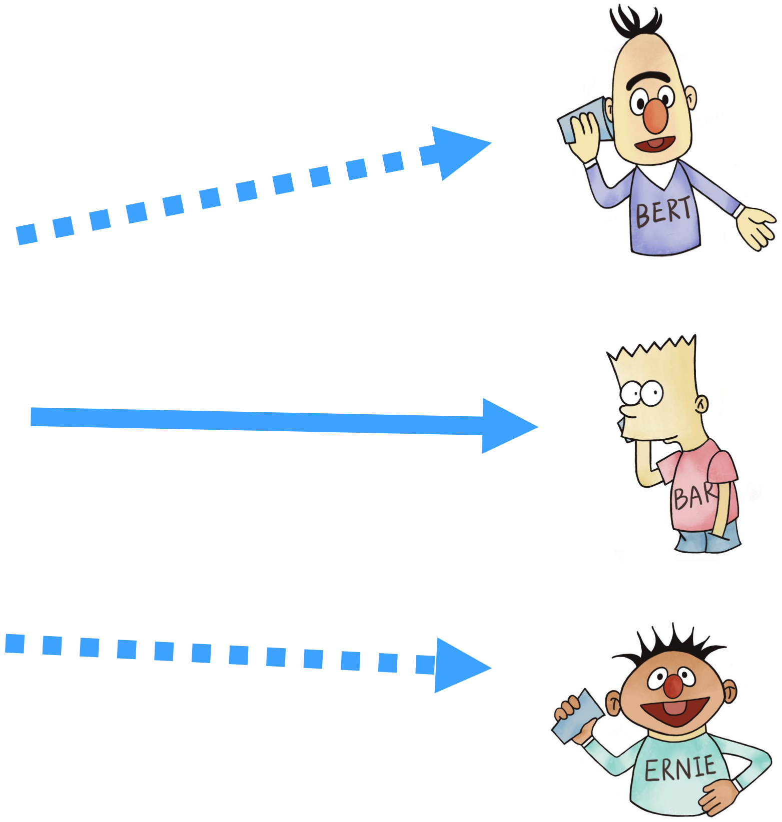
Design Considerations for Prompt-based Methods

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

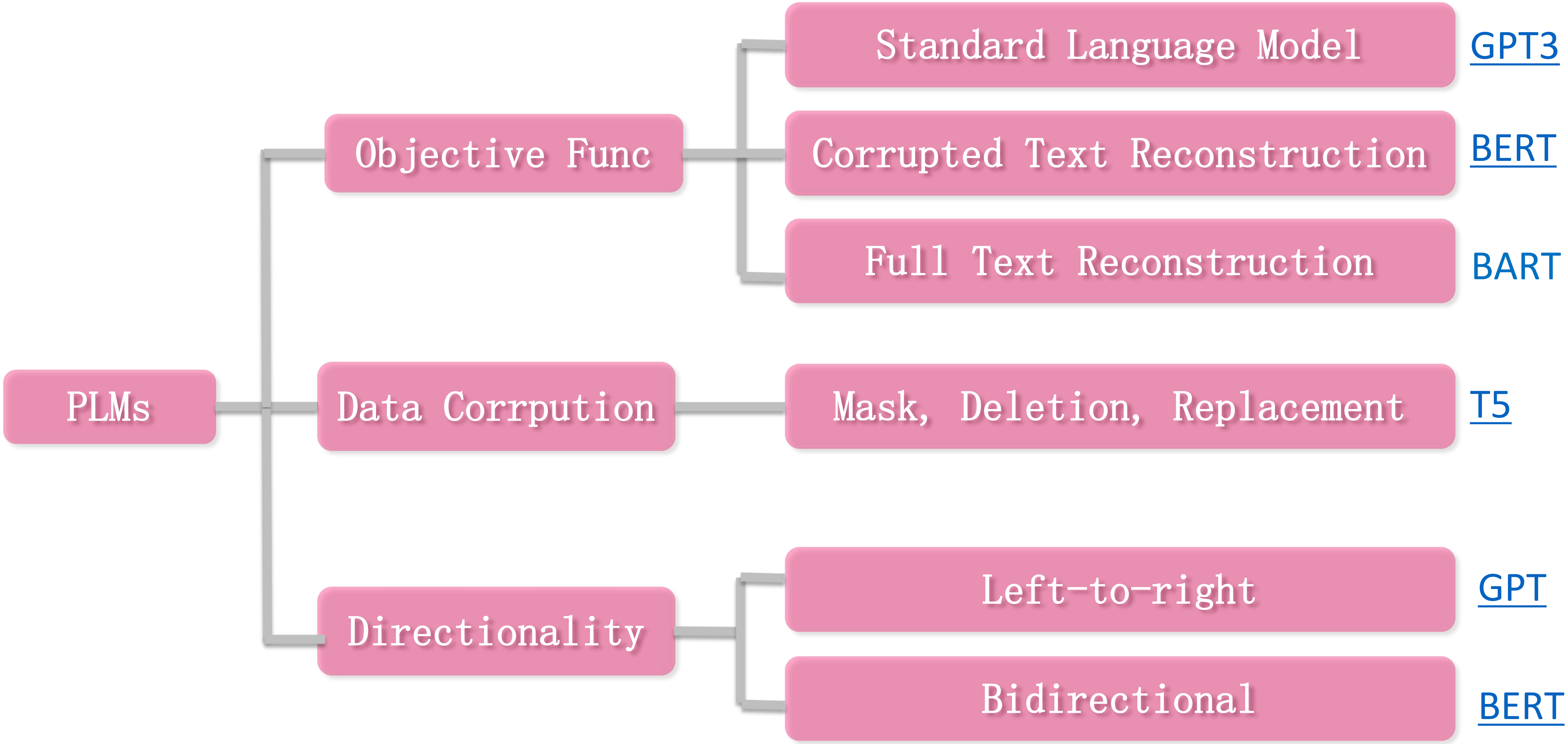
Pre-trained Model Choice

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

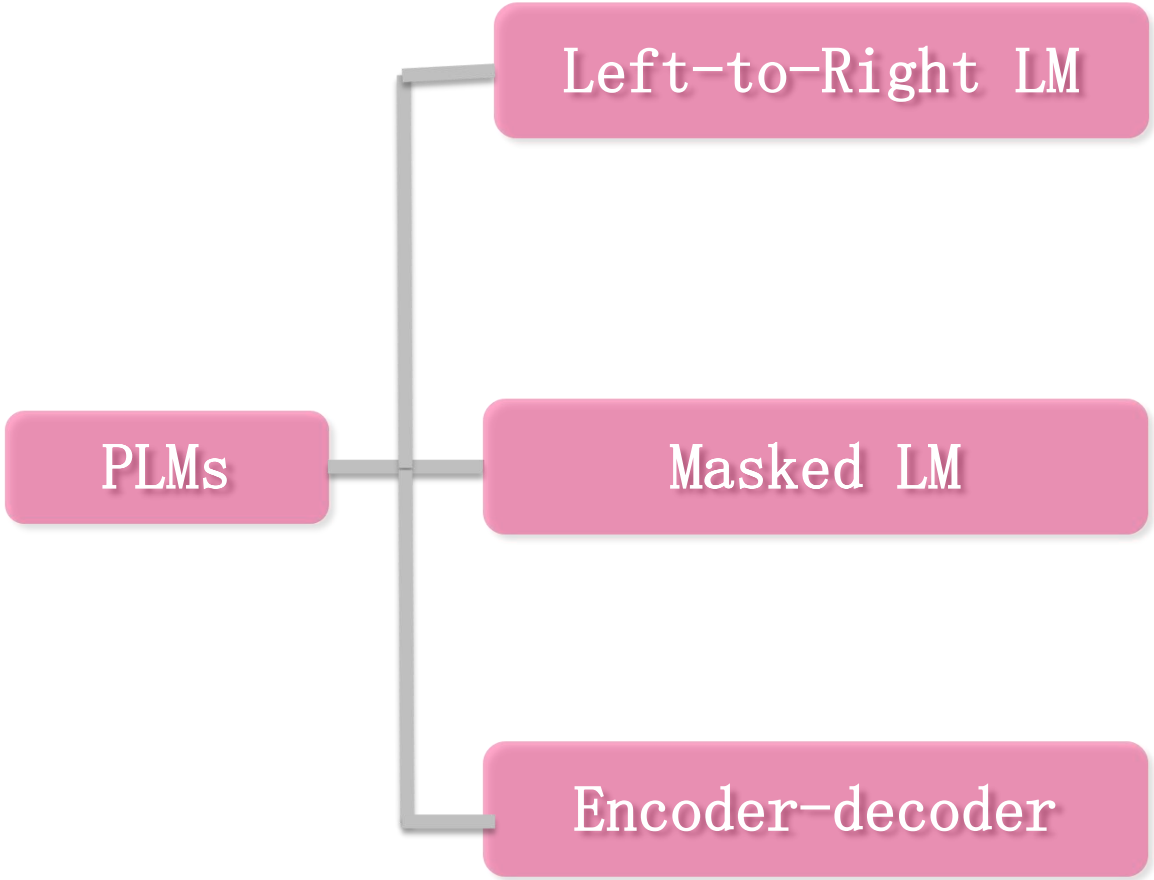
The story
describes ….,
in summary [z]



Design Decision of Pre-trained Models



Design Decision of Pre-trained Models



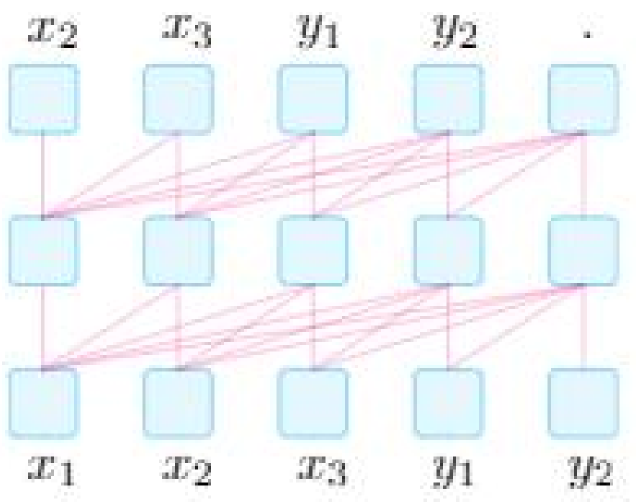
GPT, GPT3

BERT, RoBERT

BART, T5

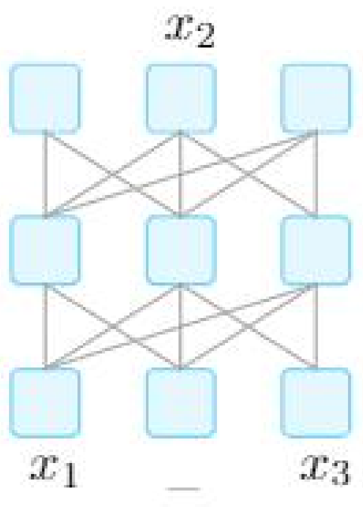
Left-to-right Language Model

- 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



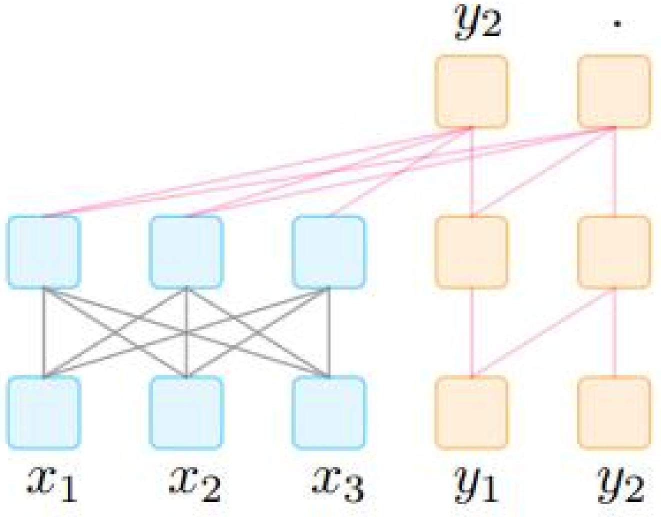
Masked Language Model

- 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

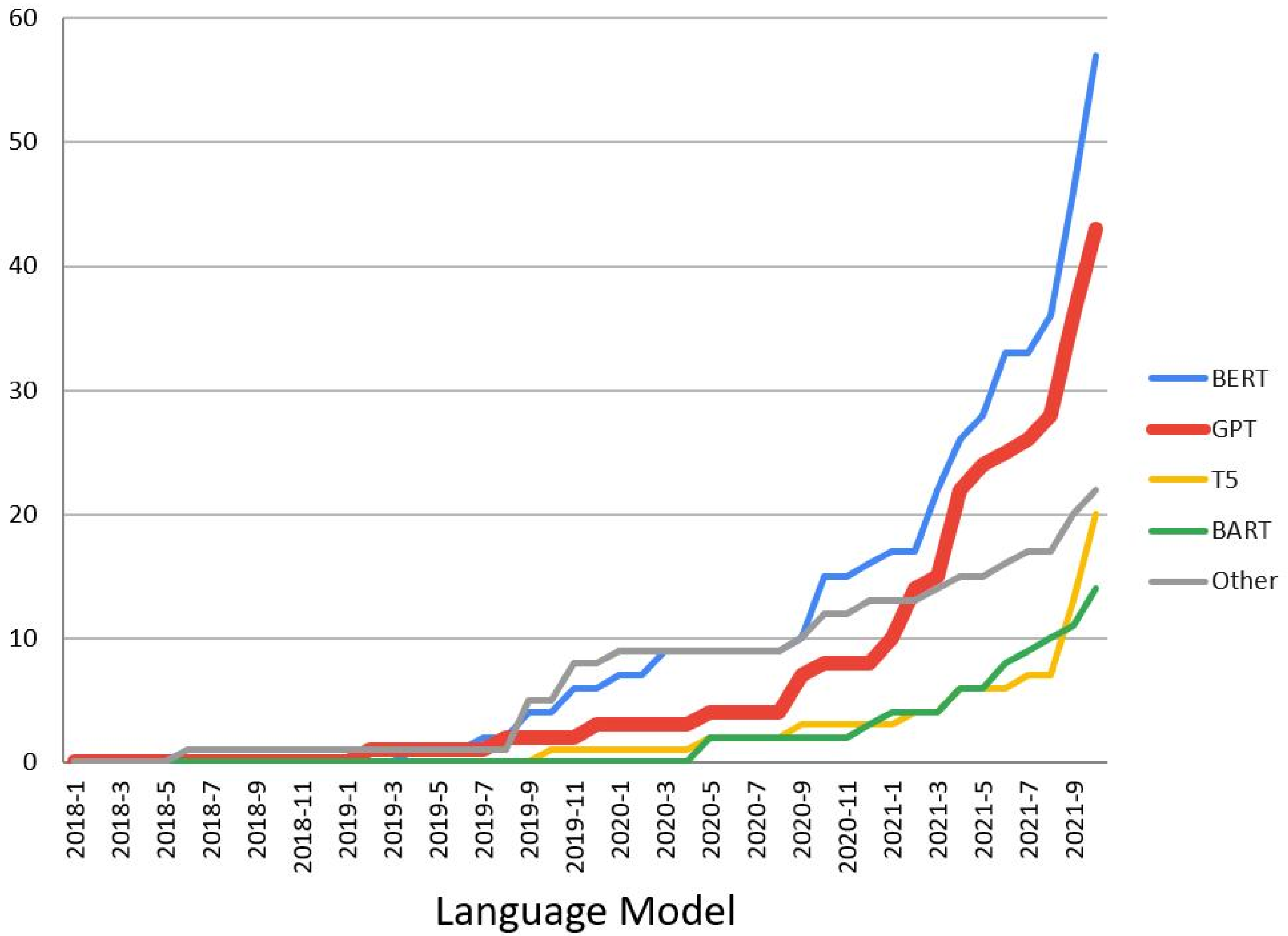


Masked Language Model

- 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



Which one is more popular?



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

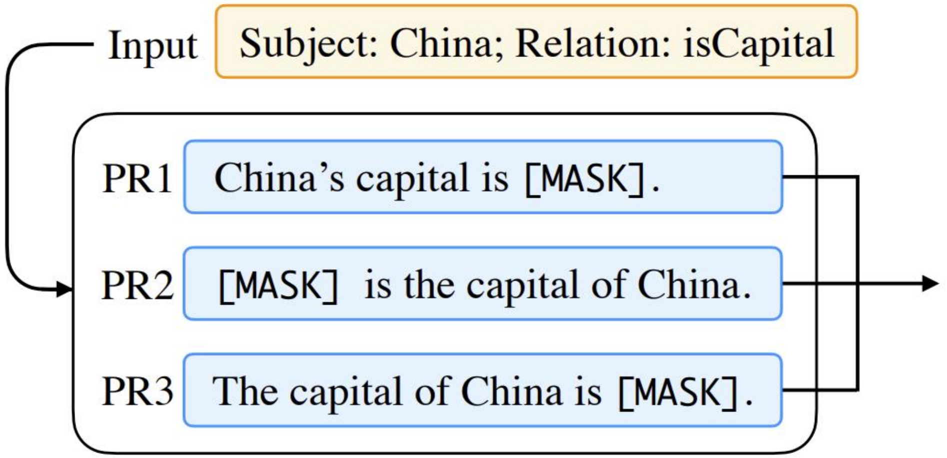
- Research Questions
  - How to extend the current prompting framework to support more NLP tasks?

Design Decision of Multiple Prompt Learning

- Multi-Prompt
  - Prompt Ensemble [11], [57]
  - In-context Learning [47], [48]
  - Chain-of-thought [52]

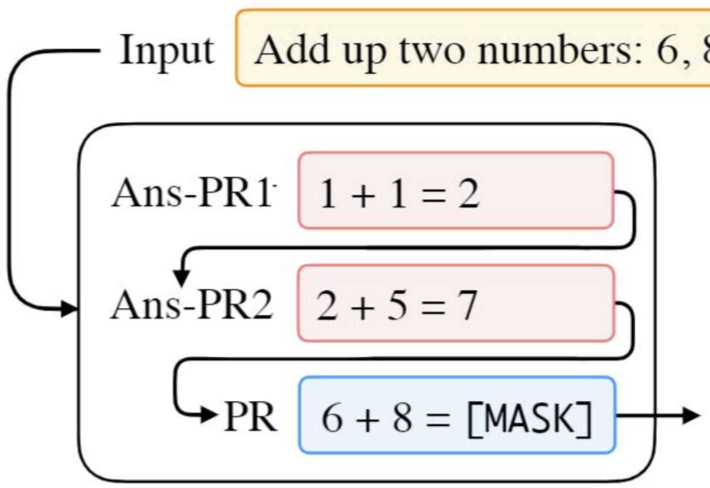
Prompt Ensembling

- **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

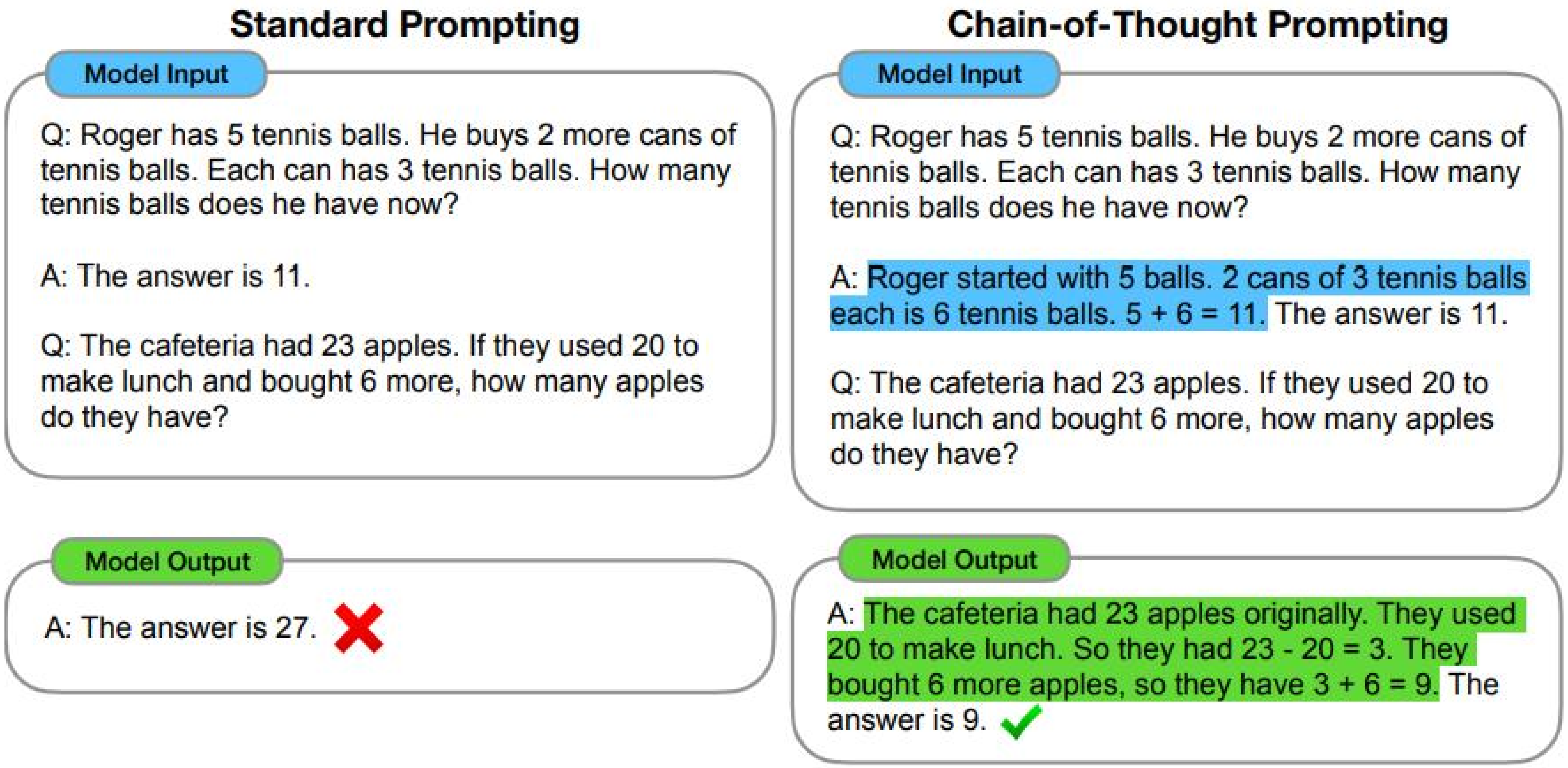


In-context Learning

- 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



Chain-of-thought



Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al.2022

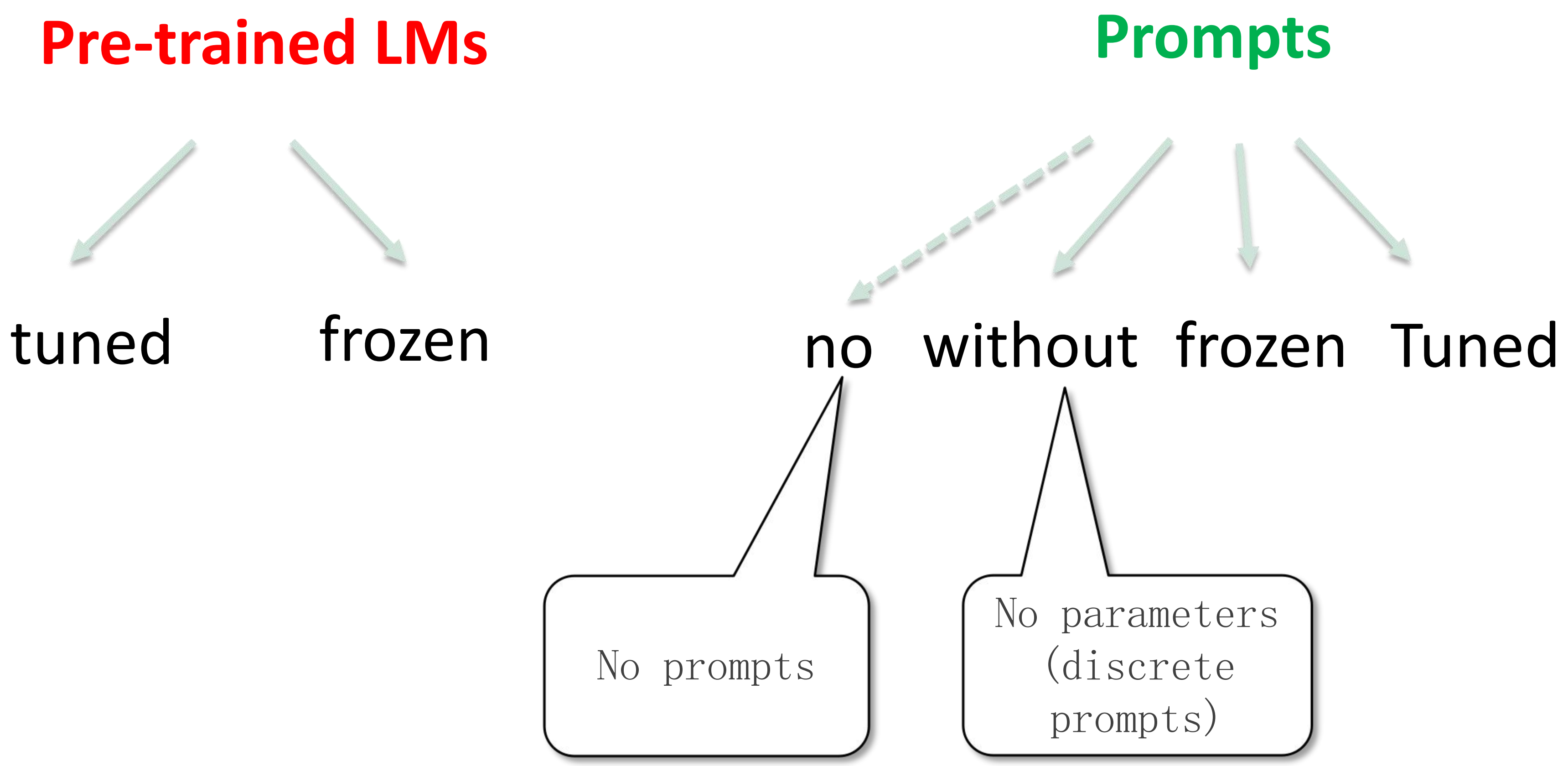
Training Strategies

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

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



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

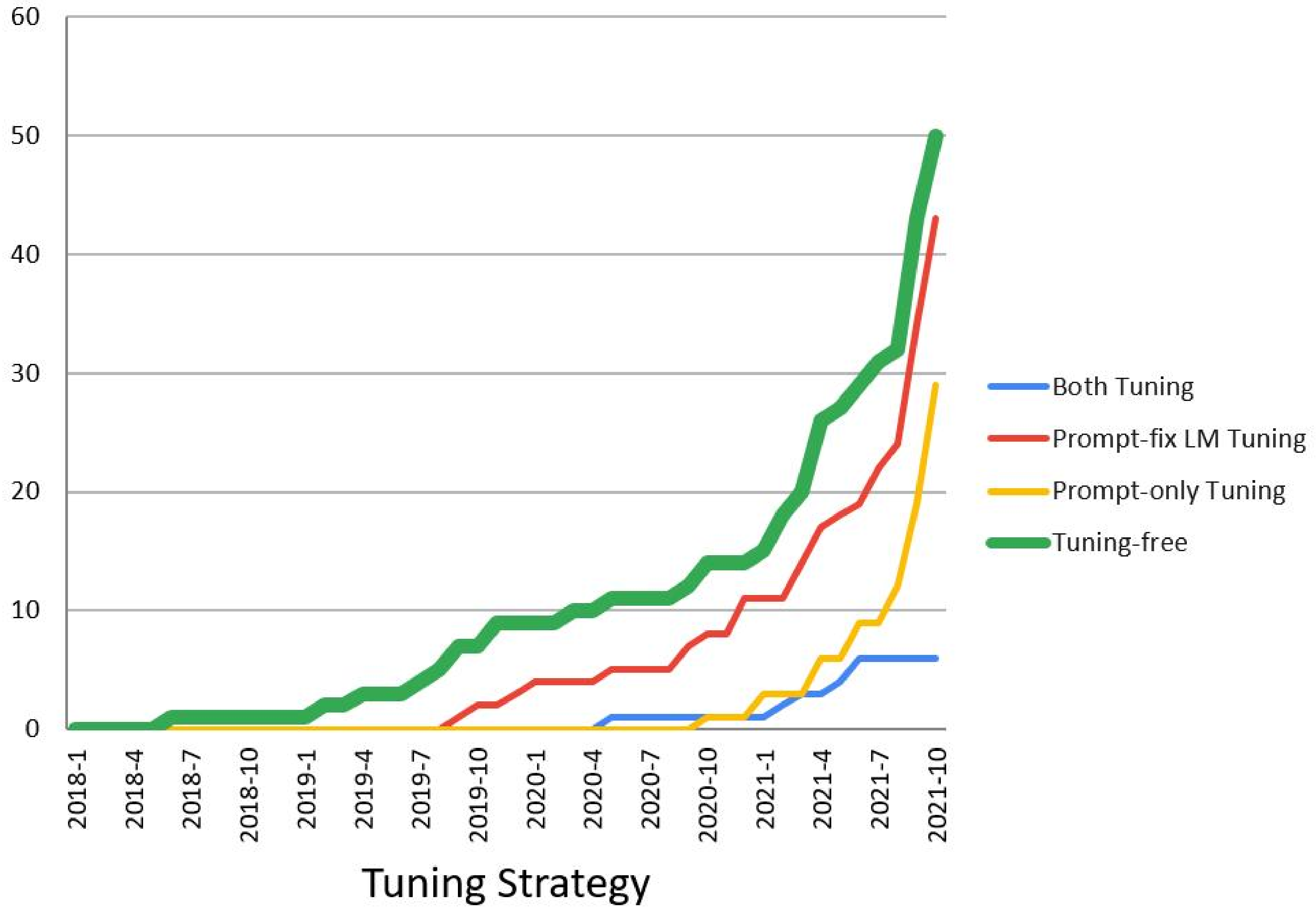
Fixed-LM Prompt Tuning

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?

Which one is more popular?

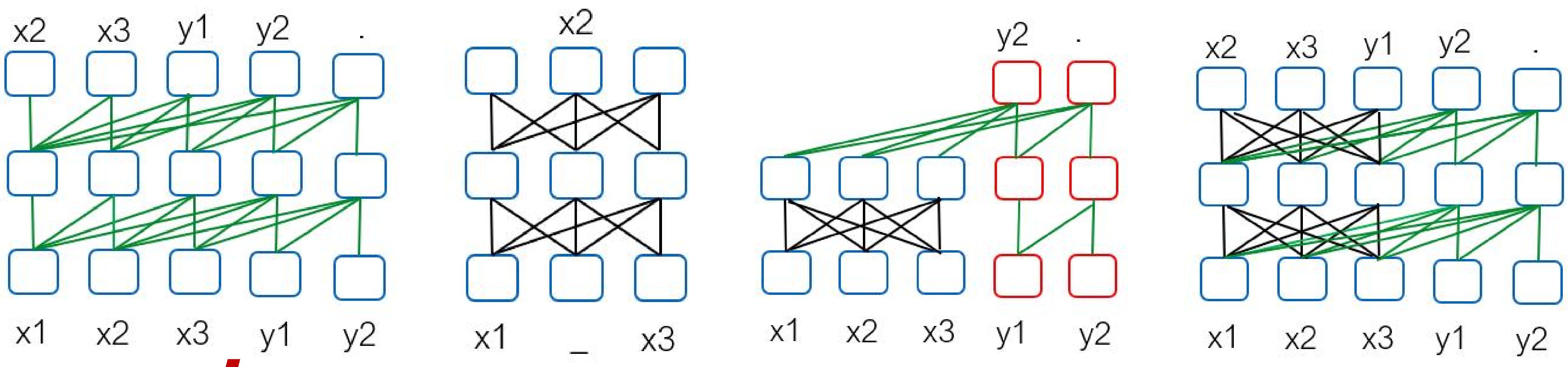


Revisit “Prompt Engineering” in the era of ChatGPT

Changes brought by ChatGPT

- Left-to-right models dominate the world

Cloze prompts fade into history

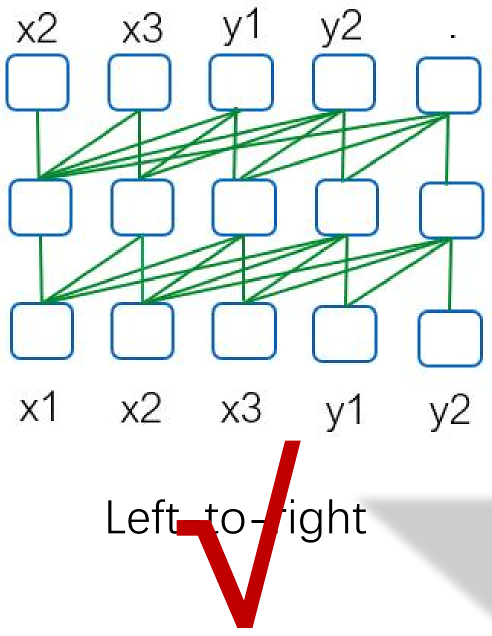


Left-to-right √ Masked LM Encode-decoder Prefixed LM

Changes brought by ChatGPT

- Left-to-right models dominate the world

Cloze prompts fade into history

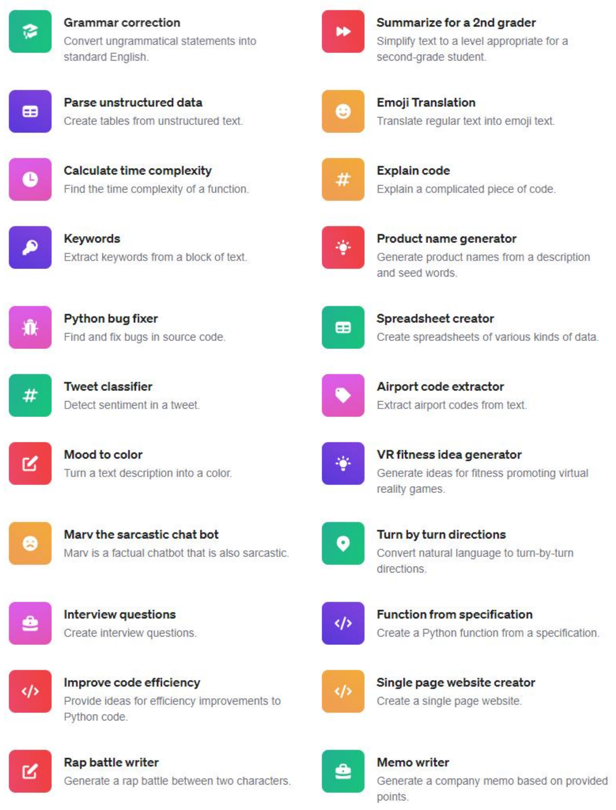


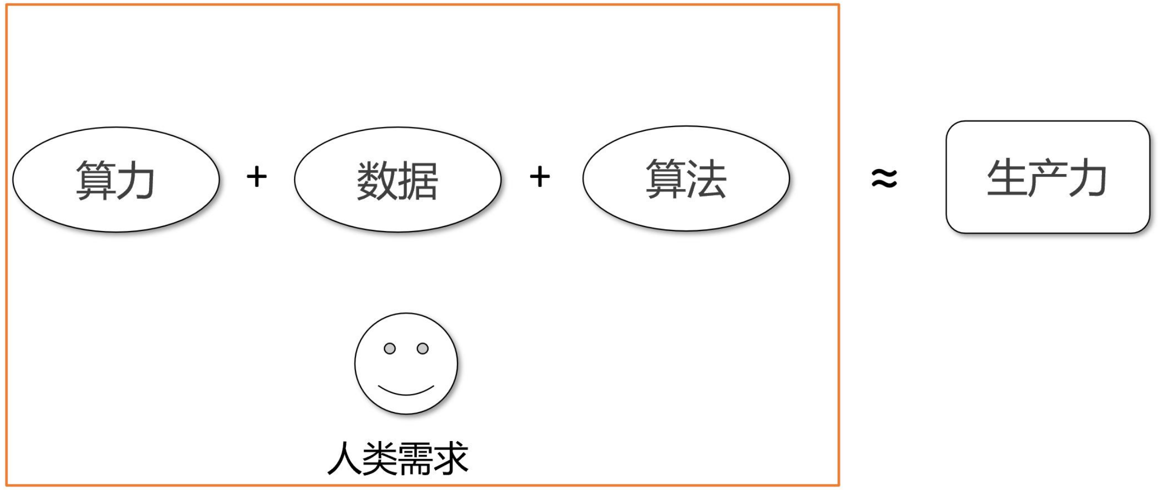
Changes brought by ChatGPT

- Left-to-right models dominate the world
- Solving traditional NLP tasks are not the most important things

Cloze prompts fade into history

Prompt distribution matters a lot





Changes brought by ChatGPT

- Left-to-right models dominate the world
- Solving traditional NLP tasks are not the most important things
- API-based research become more popular

Cloze prompts fade into history

Prompt distribution matters a lot

Zero-shot & few-shot prompting

Changes brought by ChatGPT

- Left-to-right models dominate the world
- Solving traditional NLP tasks are not the most important things
- API-based research become more popular
- Supervised fine-tuning become popular

Cloze prompts fade into history

Prompt distribution matters a lot

Zero-shot & few-shot prompting

Prompt scaling law

Changes brought by ChatGPT

- Left-to-right models dominate the world
- Solving traditional NLP tasks are not the most important things
- API-based research become more popular
- Supervised fine-tuning become popular
- Evaluation is difficult

Cloze prompts fade into history

Prompt distribution matters a lot

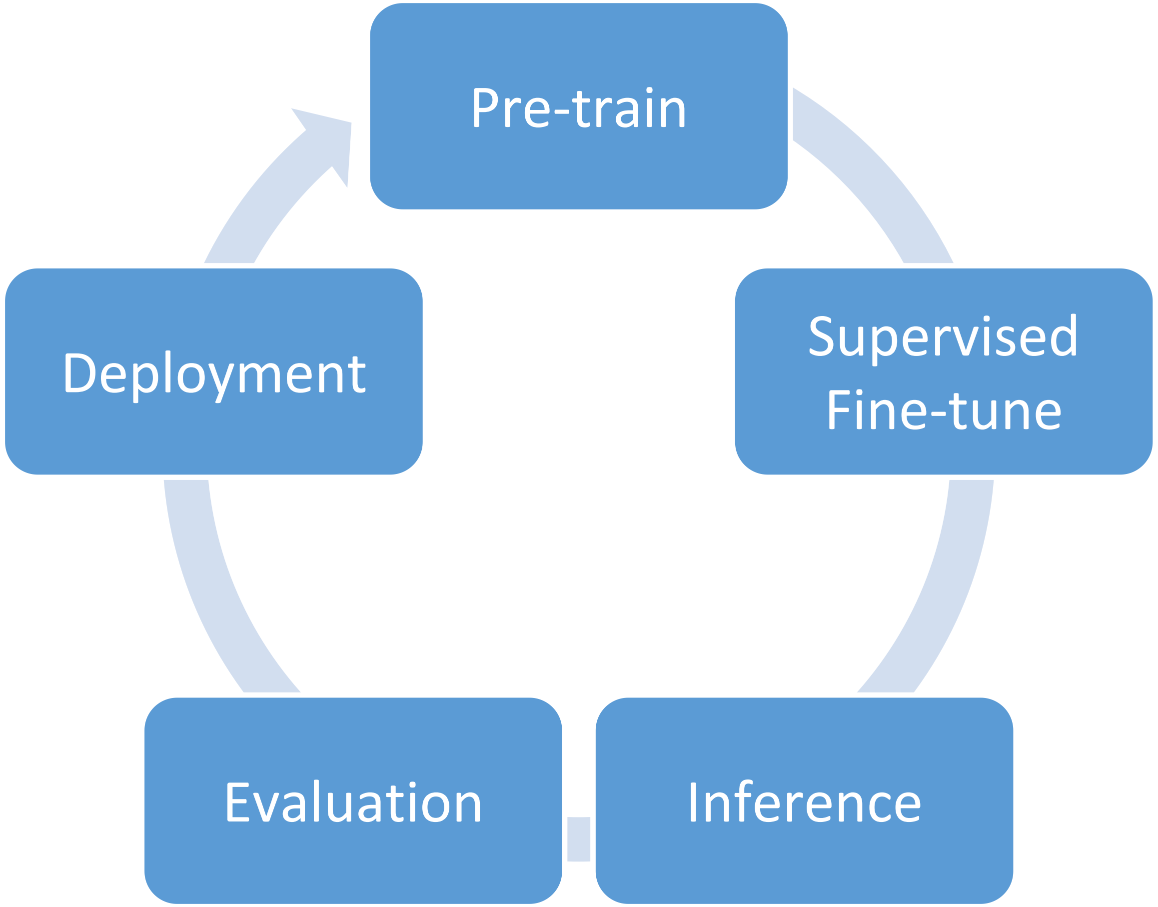
Zero-shot & few-shot prompting

Prompt scaling law

Prompt-based evaluation

Prompt Engineering 2.0: Design Considerations

Prompt Engineering in LLMOps



Prompt Engineering: Supervised Fine-tuning

- Prompt Diversity
  - How does prompt diversity affect model’s performance?
- Prompt number
  - How does the number of prompts affect model’s performance?
- Response Quality
  - How does the quality of response affect model’s performance?

Prompt Engineering: Supervised Fine-tuning

Table 3: English Instruction Data (Continued from Table 2)

Dataset	# Tasks	# Instructions	Lan	Collection Method	Usage	Access	Human Verified?
OIG (AI, 2021)	30	43M	English	Mixed	Instruct. Tuning	Open	No
Baize (Xu et al., 2023)	3	100K+	English	Model Generated	Chat	Open	No
Camel (Guohao et al., 2023)	-	115K	English	Model Generated	Instruct. Tuning, Chat	Open	No
UltraChat (Ding et al., 2023)	-	675K	English	Model Generated	Chat	Open	No
Dolly (Databricks, 2022)	7	15,000	English	Human Annotated	Instruct. Tuning	Open	Yes
Guanaco-Dataset (JosephusCheung, 2021)	175	534,530	Multilingual	Mixed	Instruct. Tuning	Open	No
ChatLLaMA Chinese-ChatLLaMA (YDli-ai, 2021)	-	-	Multilingual	Mixed	Instruct. Tuning	Open	No
GPT-4-LLM (Peng et al., 2023)	175	165K	Multilingual	Model Generated	RLHF, Instruct. Tuning	Open	No
ShareGPT (ShareGPT, 2021)	-	-	Multilingual	Model Generated	Instruct. Tuning, Chat	Closed	Yes
SHP (Ethayarajh et al., 2023)	18	385K	English	Existing, Human Annotated	RLHF, Instruct. Tuning	Open	Yes
HH-RLHF (Bai et al., 2022; Anthropic, 2022; Ganguli et al., 2022)	-	169,550	English	Mixed	RLHF, Instruct. Tuning	Open	Yes
HC3 (Guo et al., 2023)	12	37,175	Multilingual	Mixed	Instruct. Tuning	Open	Yes

A Survey of Recently Released "Instructions" (Zhang et al)

Prompt Engineering: Supervised Fine-tuning

	MMLU (factuality)	GSM (reasoning)	BBH (reasoning)	TydiQA (multilinguality)	Codex-Eval (coding)	AlpacaFarm (open-ended)	Average
	EM (0-shot)	EM (8-shot, CoT)	EM (3-shot, CoT)	F1 (1-shot, GP)	P@10 (0-shot)	Win % vs Davinci-003	
Vanilla LLaMa 13B	42.5	14.0	36.9	47.4	26.6	-	-
+SuperNI	49.8	4.0	2.8	51.4	13.1	5.0	21.0
+CoT	44.5	**39.5**	39.0	**52.2**	23.3	4.7	33.9
+Flan V2	**50.7**	21.0	39.2	47.5	16.2	5.3	30.0
+Dolly	45.3	17.0	26.0	46.8	31.4	18.3	30.8
+Open Assistant 1	43.1	16.0	38.5	38.3	31.8	55.2	37.1
+Self-instruct	30.3	9.0	29.6	40.4	13.4	7.3	21.7
+Unnatural Instructions	46.2	7.5	32.8	39.3	24.8	10.8	26.9
+Alpaca	45.1	8.0	34.5	32.8	27.6	33.2	30.2
+Code-Alpaca	42.6	12.0	36.6	41.3	34.5	21.3	31.4
+GPT4-Alpaca	47.0	14.0	38.3	24.4	32.5	63.6	36.6
+Baize	43.5	8.5	36.7	33.9	27.3	33.9	30.6
+ShareGPT	49.2	16.0	40.1	30.1	31.6	**69.1**	39.3
+ Human data mix	50.4	36.5	39.4	49.8	23.7	38.5	39.7
+Human+GPT data mix.	49.2	36.5	**42.8**	46.1	**35.0**	57.2	**44.5**

Which “instruction” data is the best? (Wang et al)

Prompt Engineering: Supervised Fine-tuning

Source	#Examples	Avg Input Len.	Avg Output Len.
Training			
Stack Exchange (STEM)	200	117	523
Stack Exchange (Other)	200	119	530
wikiHow	200	12	1,811
Pushshift r/WritingPrompts	150	34	274
Natural Instructions	50	236	92
Paper Authors (Group A)	200	40	334
Dev			
Paper Authors (Group A)	50	36	N/A
Test			
Pushshift r/AskReddit	70	30	N/A
Paper Authors (Group B)	230	31	N/A

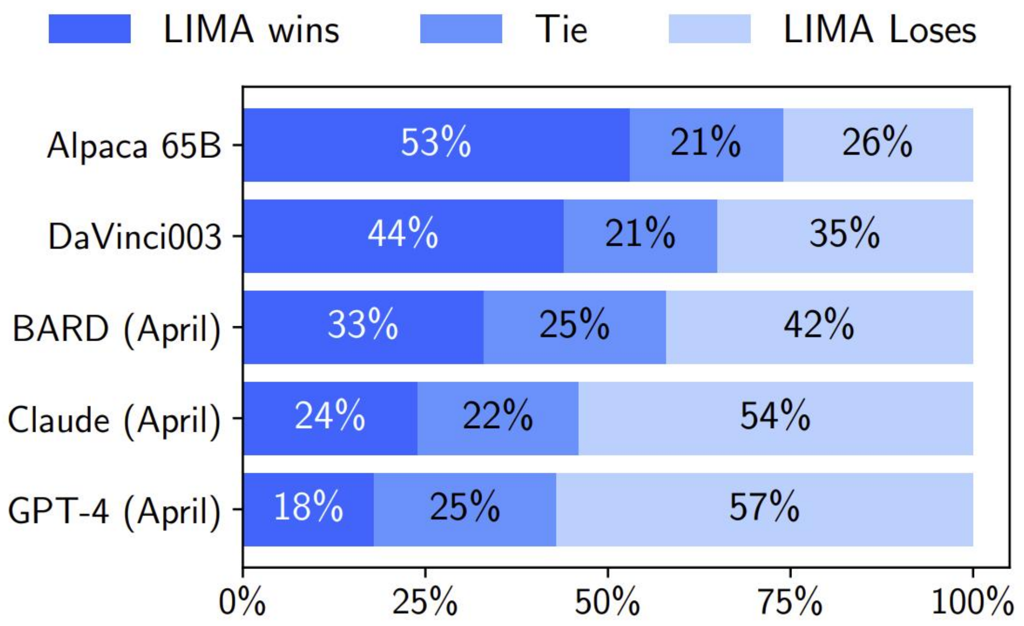


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

LIMA: Less Is More for Alignment (Zhou et al)

Prompt Engineering: Inference

- Zero-shot Prompting:
  - How to ask a good question that ChatGPT can better understand you?

Prompt Engineering: Inference

你是一个中文人工智能助手，你需要仿照示例，根据给定的除示例外的所有法律生成一个包含题目、选项分析和答案的单项选择题。在生成单项选择题时，你必须遵守以下几个原则：

题目构成 1. 题目由题目描述和4个选项构成

题目描述 2. 单项选择题的题目描述需要合理

题目生成的整体限制 3. 尽可能根据除示例外的所有法律生成题目，避免使用单条法律生成题目

题目选项
4. 在生成4个选项时，结合题目描述与除示例外的所有法律，首先设计1个正确答案的选项，然后再设计3个错误的选项，接着这4个选项以随机的顺序排列
5. 选项互有差异，避免选项之间的明显重复或相似性
6. 在设计选项时，不要使得某些选项明显不可能是正确答案
7. 每个选项需要和题目描述相关
8. 每个选项需要前后内容一致
9. 不能直接从给定的法律中复制文本作为选项内容，需要结合给定的法律生成合理的选项

生成顺序 10. 依次生成题目、选项分析和答案

选项分析 11. 选项分析是结合题目与除示例外的所有法律，对每个选项进行分析

答案 12. 选项分析中的正确答案是最终答案

以下是1个示例：
示例：
{example}

让我们一步一步思考，参考示例并结合给定法律"{input_law}"{action}，依次生成下面内容：

题目：

选项分析：

答案：

法律：企业破产法：第四十六条　未到期的债权，在破产申请受理时视为到期。附利息的债权自破产申请受理时起停止计息。第四十七条　附条件、附期限的债权
题目：A公司因经营不善，资产已不足以清偿全部债务，经申请进入破产还债程序。关于破产债权的申报，下列哪个表述是正确的?
A.甲对A公司的债权虽未到期，不可以申报
B.乙对A公司的债权因附有条件，故不能申报
C.丙对A公司的债权虽然诉讼未决，但丙仍可以申报
D.职工丁对A公司的伤残补助请求权，应予以申报
选项分析：《企业破产法》第46条第一款规定，未到期的债权，在破产申请受理时视为到期。据此可知，未到期的债权，仍可申报。选项A错误。《企业破产法》
答案：C

中华人民共和国河道管理条例规定：第十条　河道的整治与建设，应当服从流域综合规划，符合国家规定的防洪标准、通航标准和其他有关技术要求，维护堤防安全，保持河势稳定和行洪、航运通畅。第十一条　修建开发水利……

设计一个法律情景/针对给定法律中的某个概念

Prompt Engineering: Changes brought by ChatGPT

- Zero-shot Prompting
- Few-shot Prompting
  - How do I get the model to mimic a given example?
    - Format following
    - Reasoning step decomposition

“X”- of thought

Chain-of-thought

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 tennis balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves of bread did they have left?

Model Output

A: The bakers started with 200 loaves. They sold 93 in the morning and 39 in the afternoon. So they sold 93 + 39 = 132 loaves. The grocery store returned 6 loaves. So they had 200 - 132 - 6 = 62 loaves left.
The answer is 62.

Program-of-thought

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 tennis balls.
`tennis_balls = 5`
2 cans of 3 tennis balls each is
`bought_balls = 2 * 3`
tennis balls. The answer is
`answer = tennis_balls + bought_balls`

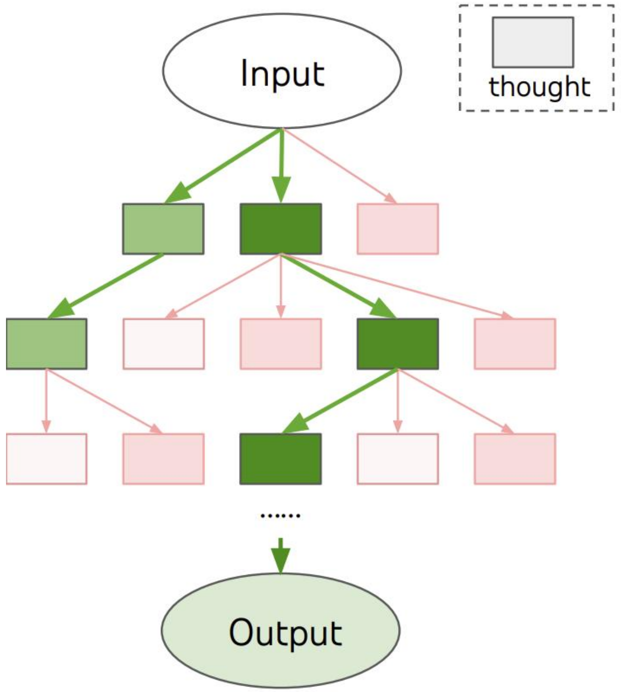
Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves of bread did they have left?

Model Output

A: The bakers started with 200 loaves
`loaves_baked = 200`
They sold 93 in the morning and 39 in the afternoon
`loaves_sold_morning = 93`
`loaves_sold_afternoon = 39`
The grocery store returned 6 loaves.
`loaves_returned = 6`
The answer is
```
answer = loaves_baked - loaves_sold_morning
  - loaves_sold_afternoon + loaves_returned
```

```
>>> print(answer)
74
```

Tree-of-thought



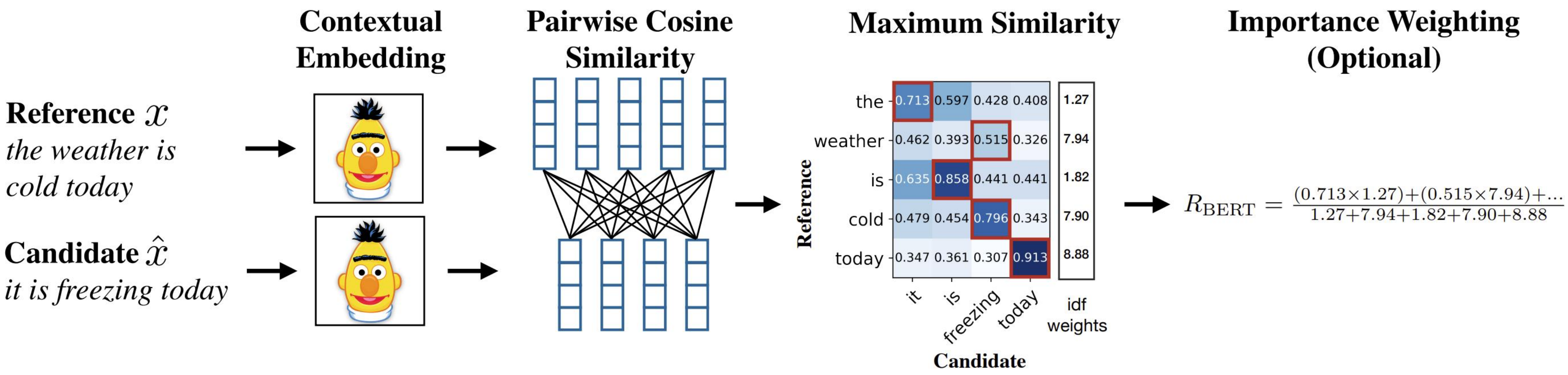
Prompt Engineering: Evaluation

- How to evaluate a model as you desire?

Prompt Engineering: Evaluation

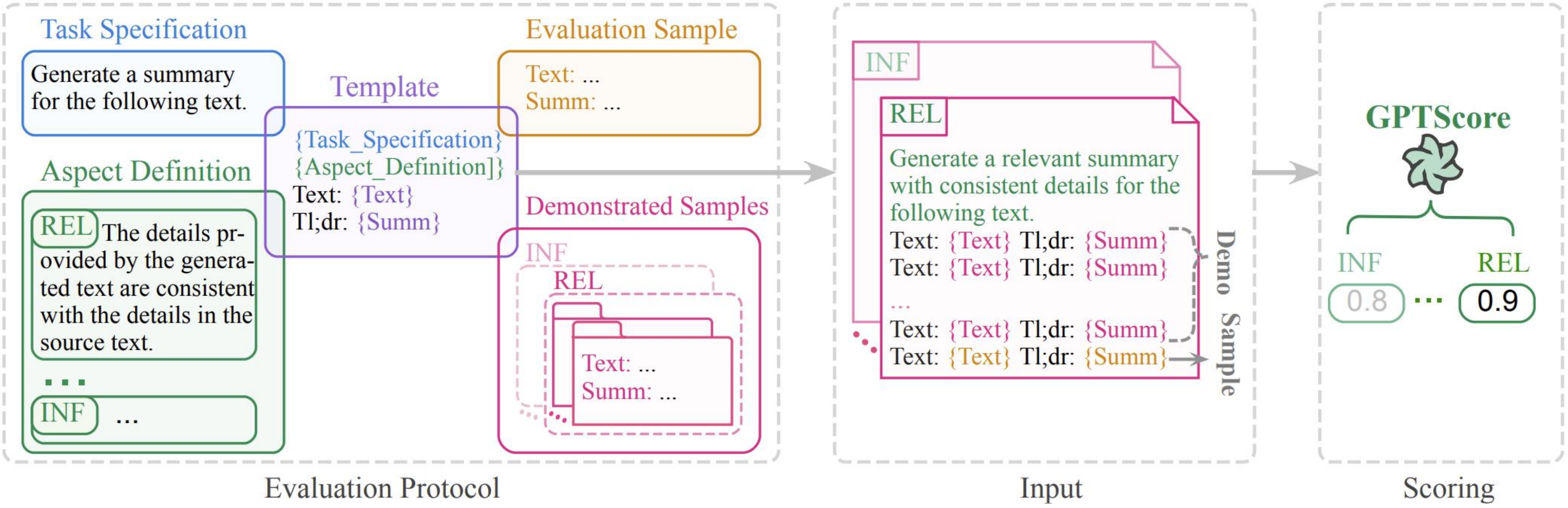
- How to evaluate a model as you desire?

BERTScore



Prompt Engineering: Evaluation

- Evaluation
  - How to evaluate a model as you desire? **GPTScore**



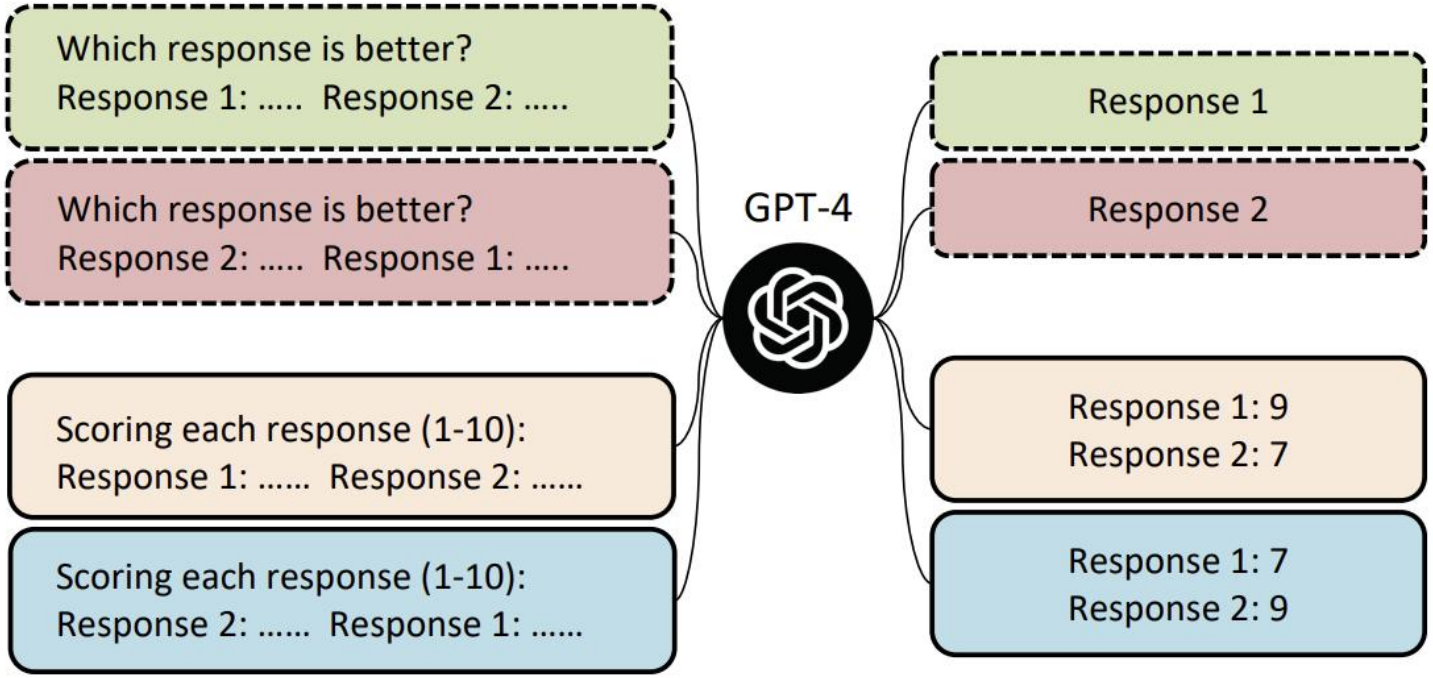
Prompt Engineering: Evaluation

- Evaluation
  - How to evaluate a model as you desire? **ChatGPT Score**

```yaml
prompt: |-
  You are evaluating a response that has been submitted for a particular task, using a specific set of standards. Below is the data:
  [BEGIN DATA]
  ***
  [Task]: {input}
  ***
  [Submission]: {completion}
  ***
  [Criterion]: {criteria}
  ***
  [END DATA]
  Does the submission meet the criterion? First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at
  Reasoning:
eval_type: cot_likert
choice_scores:
  "1": 1.0
  "2": 2.0
  "3": 3.0
  "4": 4.0
  "5": 5.0
  "6": 6.0
criteria:
  helpfulness:
    "1": "Not helpful - The generated text is completely irrelevant, unclear, or incomplete. It does not provide any useful information to the user."
    "2": "Somewhat helpful - The generated text has some relevance to the user's question, but it may be unclear or incomplete. It provides only partial information, or the information provided may not be us
    "3": "Moderately helpful - The generated text is relevant to the user's question, and it provides a clear and complete answer. However, it may lack detail or explanation that would be helpful for the use
    "4": "Helpful - The generated text is quite relevant to the user's question, and it provides a clear, complete, and detailed answer. It offers additional information or explanations that are useful for t
    "5": "Very helpful - The generated text is highly relevant to the user's question, and it provides a clear, complete, and detailed answer. It offers additional information, explanations, or analogies tha
    "6": "Highly helpful -  The generated text provides a clear, complete, and detailed answer. It offers additional information or explanations that are not only useful but also insightful and valuable to t
```

Prompt Engineering: Evaluation

- How to evaluate a model as you desire?



Prompt Engineering: Deployment

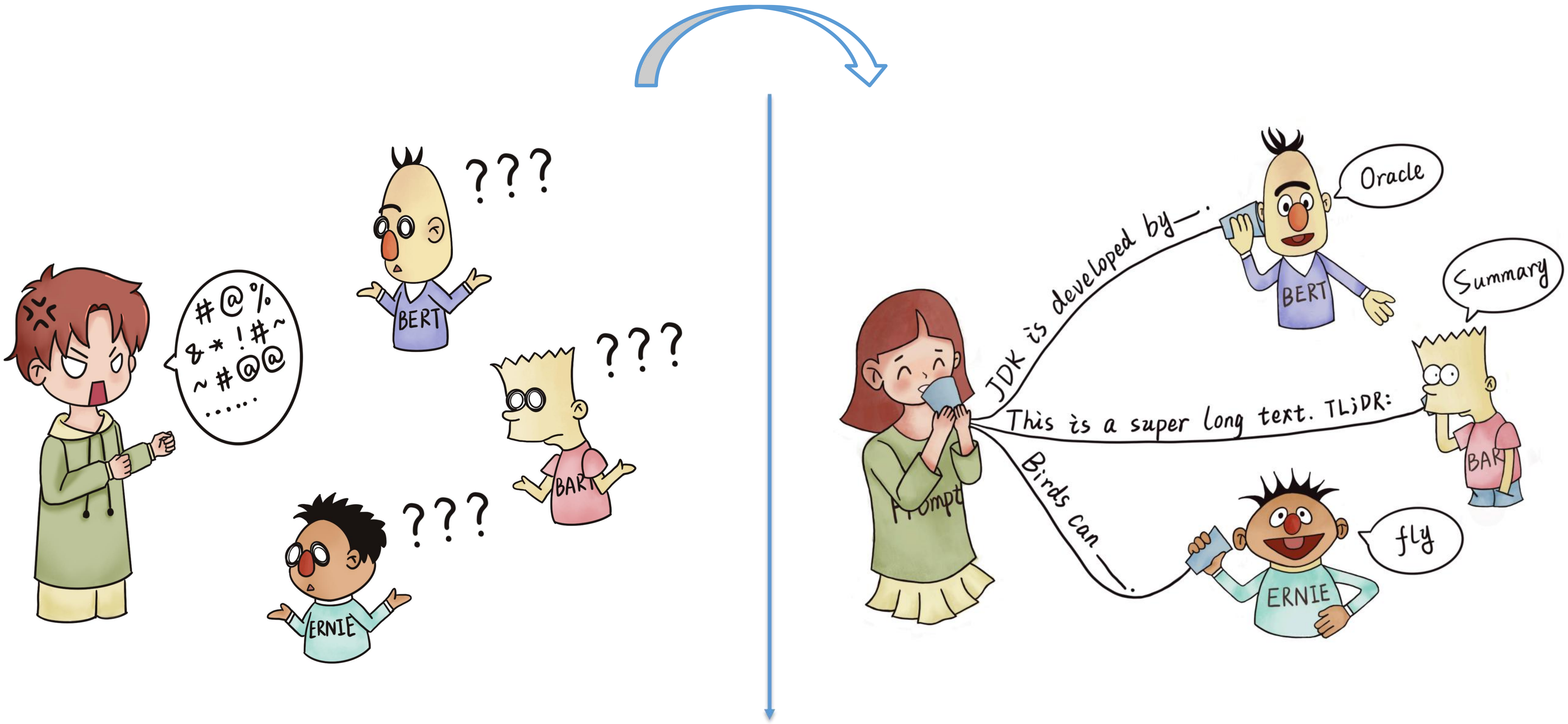
- How to design a good preface?
  - GPT Agent
  - System Message
- How to prevent jailbreak prompt?

```python
import openai

openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Who won the world series in 2020?"},
        {"role": "assistant", "content": "The Los Angeles Dodgers won the Worl
        {"role": "user", "content": "Where was it played?"}
    ]
)
```

Prompt Engineering: Pre-train

- How to prompt pre-training data so that
  - the next word could be better predicted
  - the stored information can be better elicited



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

谢谢各位！