Prompt

What is a Prompt?

prompt | präm(p)t |

verb *[with object]*

1 (of an event or fact) cause or bring about (an action or feeling): *his death has prompted an industry-wide investigation of safety violations*.

- cause (someone) to take a course of action: *a demonstration by 20,000 people prompted the government to step up security*.

2 assist or encourage (a hesitating speaker) to say something: *[with direct speech] : "And the picture?" he prompted*.

- supply a forgotten word or line to (an actor) during the performance of a play.
- *Computing* (of a computer) request input from (a user): *the online form prompts users for data*.

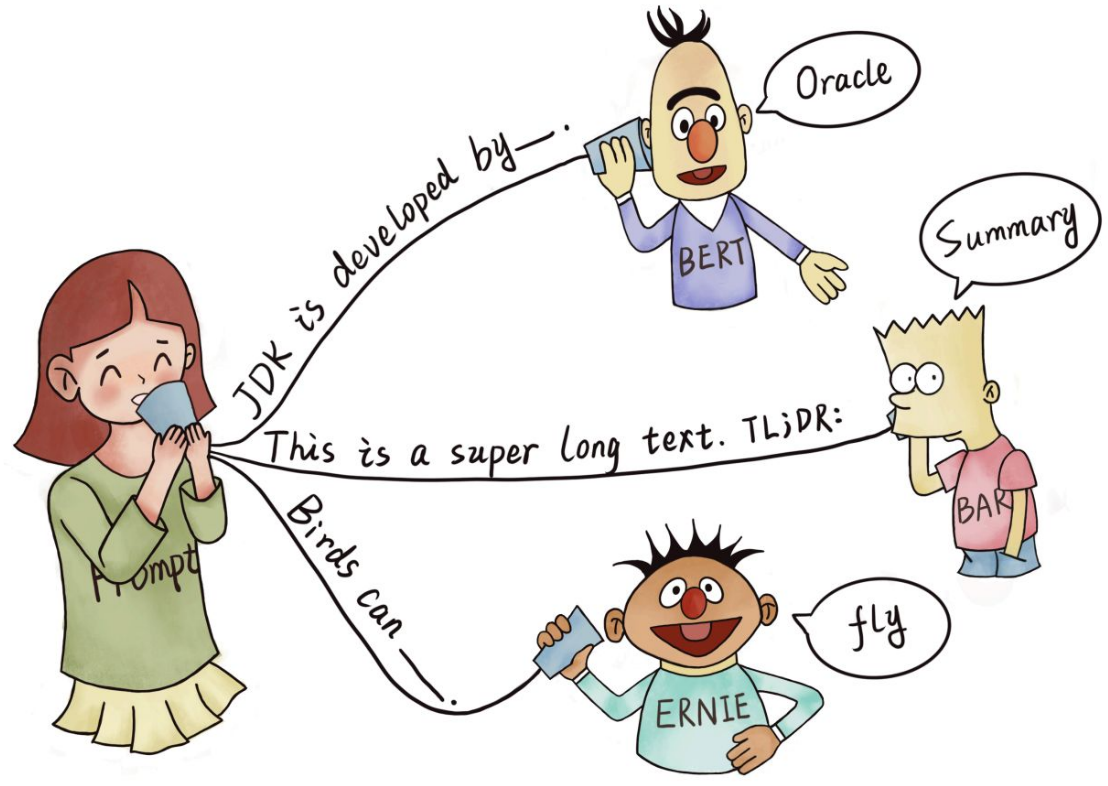
noun

1 an act of assisting or encouraging a hesitating speaker: *with barely a prompt, Barbara talked on*.

- a word or phrase spoken as a reminder to an actor of a forgotten word or line.
- another term for prompter.
- *Computing* a message or symbol on a screen to show that the system is waiting for input.

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

Prompting 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

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

METHOD

Task reformulation

Reformulating NLP tasks using prompting

1. Prompt Construction
2. Answer Construction
3. Answer Prediction
4. Answer-Label Mapping

Prompting for <u>Sentiment Classification</u>

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];

Requires 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 Prediction

Given a prompt, predict the answer [z].

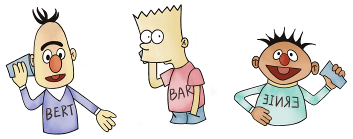
- Choose a suitable pretrained language model;

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 LM?

Step 3: Answer Prediction

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.

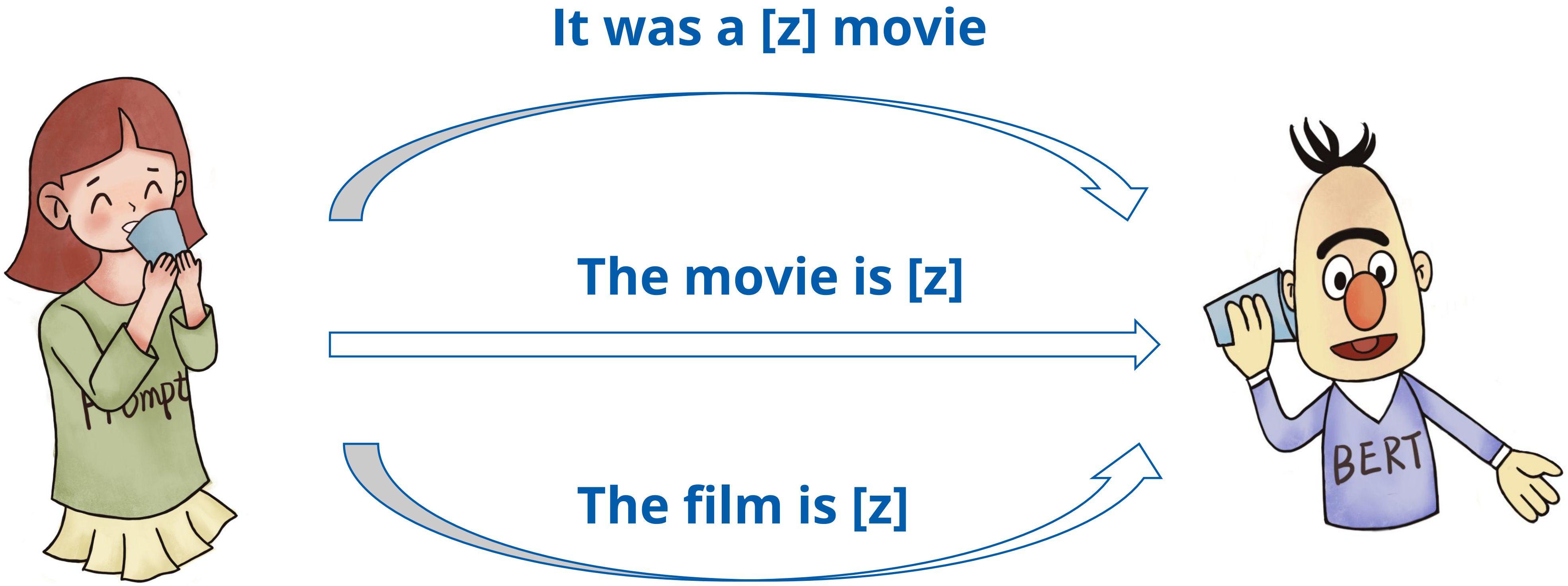
Mapping: fantastic =>☺️

Prompt Template Engineering

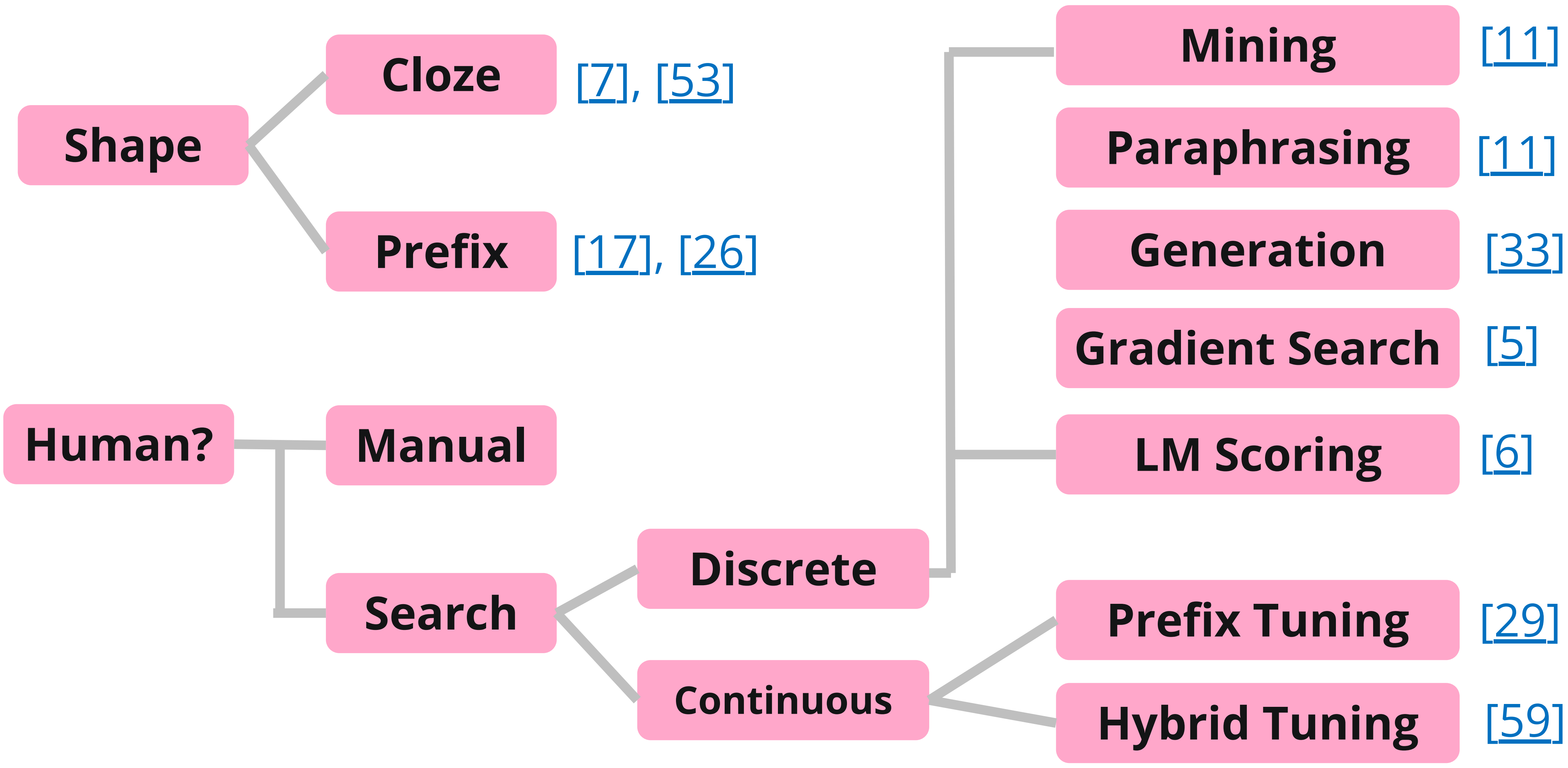
Prompt Template Engineering

Research Question:

- **How to define appropriate prompt templates**

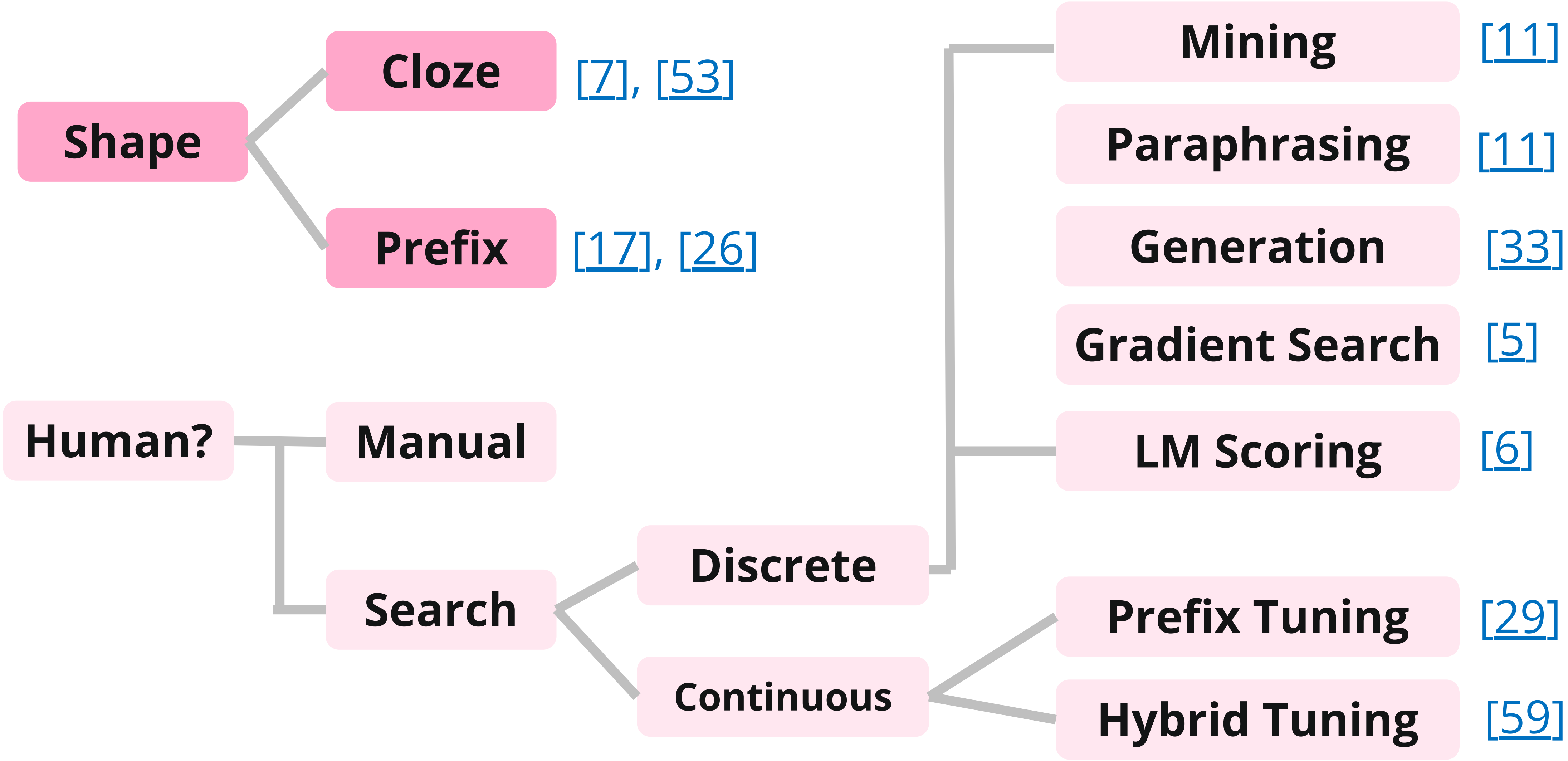


Design Decisions for Prompt Templates



Liu et al. **Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.** ACM Computing Surveys (2023).

Design Decisions for Prompt Templates



Prompt Shape

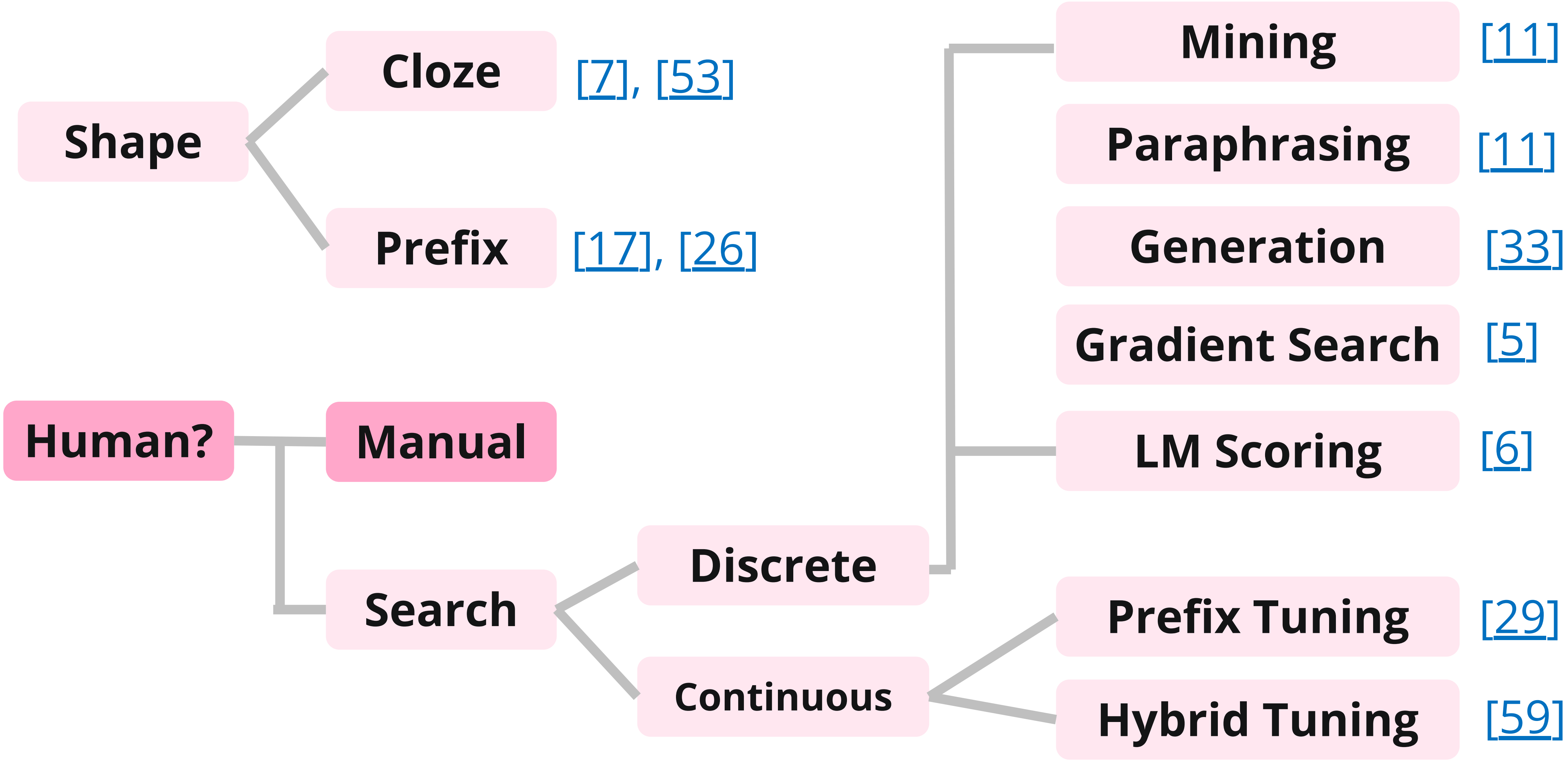
- **Cloze Template**
  - Contain blanks to be filled.
  - Useful for Masked LMs.
    - The capital of ____ is Beijing .

Prompt Shape

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

- **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: ____

Design Decision of Prompt Templates



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. 😔

- One template–answer pair

Task Accuracy

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

Answer: fantastic/terrible

75%

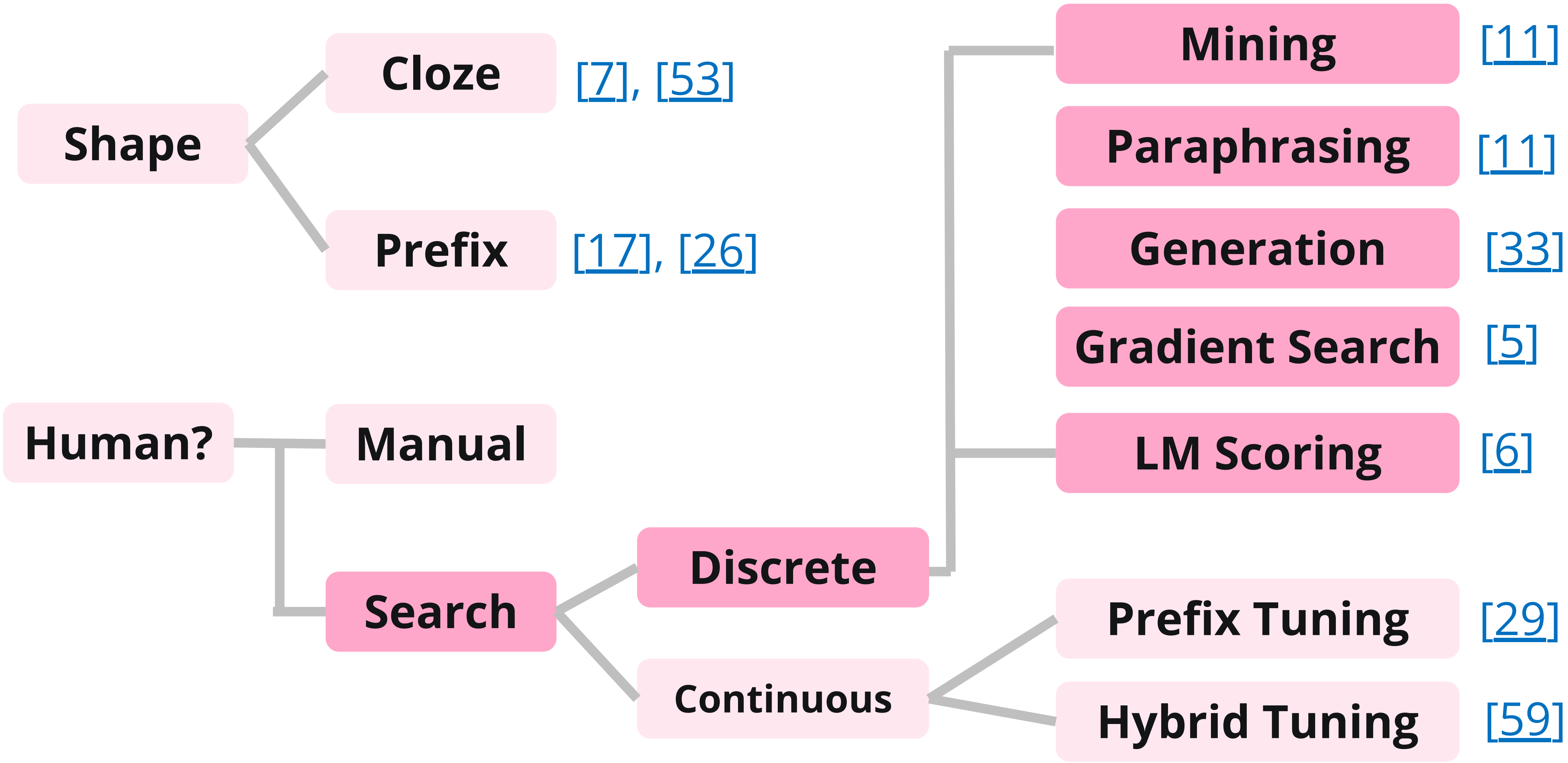
- Another template–answer pair

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

Answer: positive/negative

53%

Design Decisions for Prompt Templates



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)

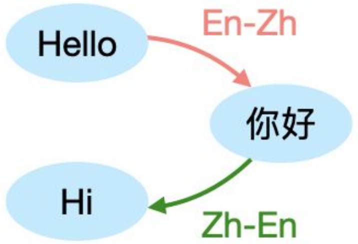
Paraphrasing

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

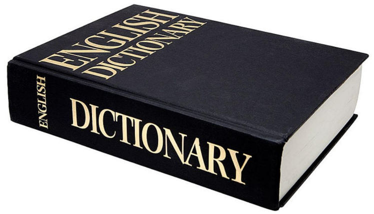
References: [1] Jiang et al. **How Can We Know What Language Models Know?** TACL (2020). [2] Yuan et al. **BARTScore: Evaluating Generated Text as Text Generation**. NeurIPS (2021). [3] Haviv et al. **BERTese: Learning to Speak to BERT.** EACL (2021).

Paraphrasing

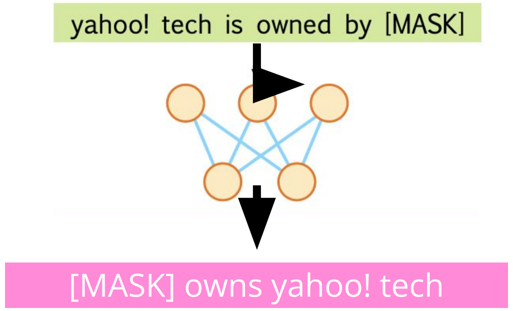
- Take in an existing seed template and paraphrase it into a set of other candidate templates.
- Typical methods
  - Back-translation



  - Using replacement of phrases from a thesaurus



  - Use neural rewriter to rewrite



Gradient-based Search

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

"I love this movie!" $\square$ ___ .

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

The **template token** we want to search

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

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
conveys	0.7
hello	0.09
cat	0.04
….	…

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
conveys	0.7
hello	0.09
cat	0.04
….	…

Generation

- Use LM to generate templates.
  - T5

Pre-train	Input: Thank you <X> me to the party <Y> week.
	Target: <X> = for inviting <Y> = last

Reference: Gao et al. **Making Pre-trained Language Models Better Few-shot Learners.** ACL (2021).

Generation

- Use LM to generate templates.
  - T5

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

T5 decode

<X> = This is a <Y> = .
<X> = I thought it was a <Y> = one .
…

LM Scoring

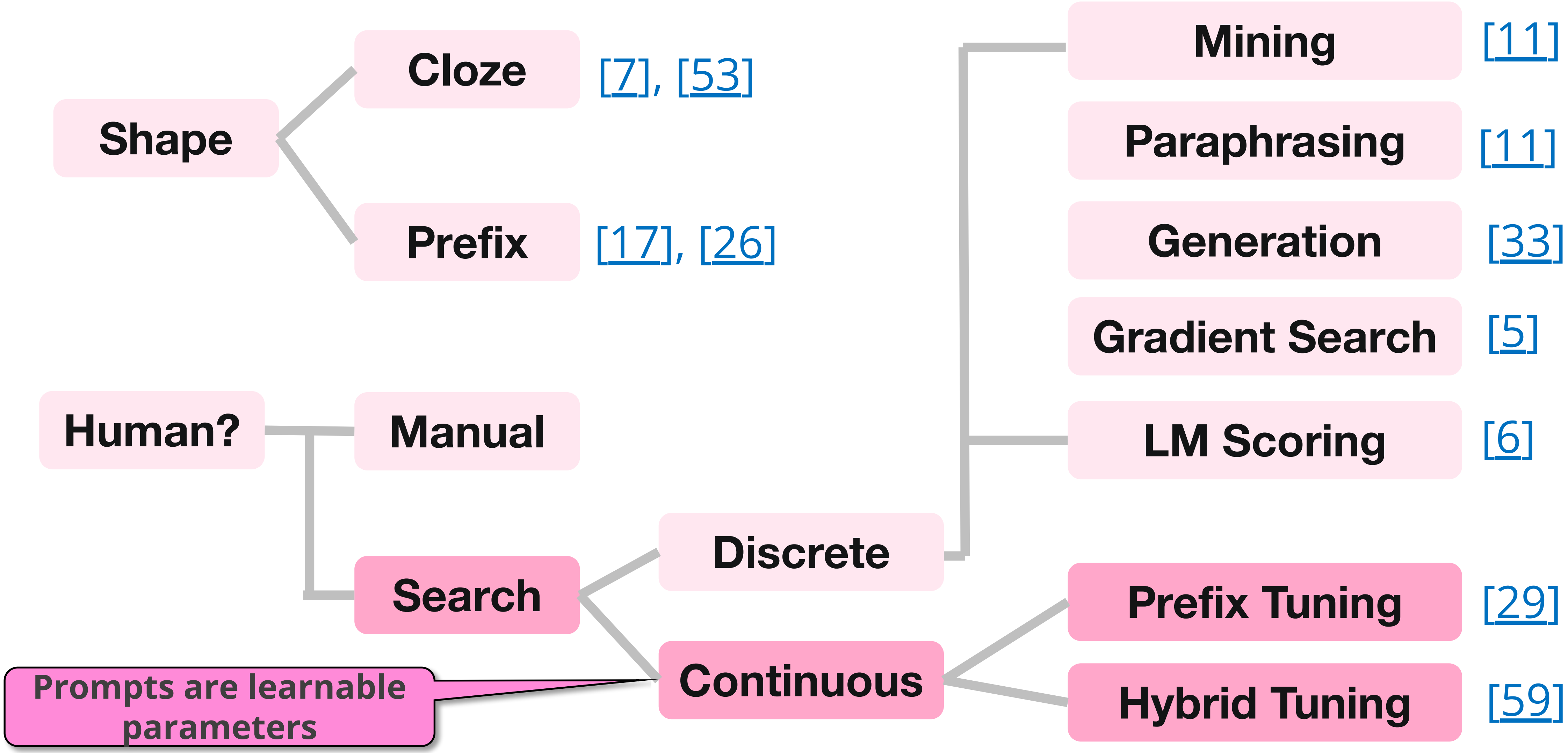
- Use the LM to choose the templates that achieve high LM probability.

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 Decisions for Prompt Templates



Answer Engineering

Answer Engineering

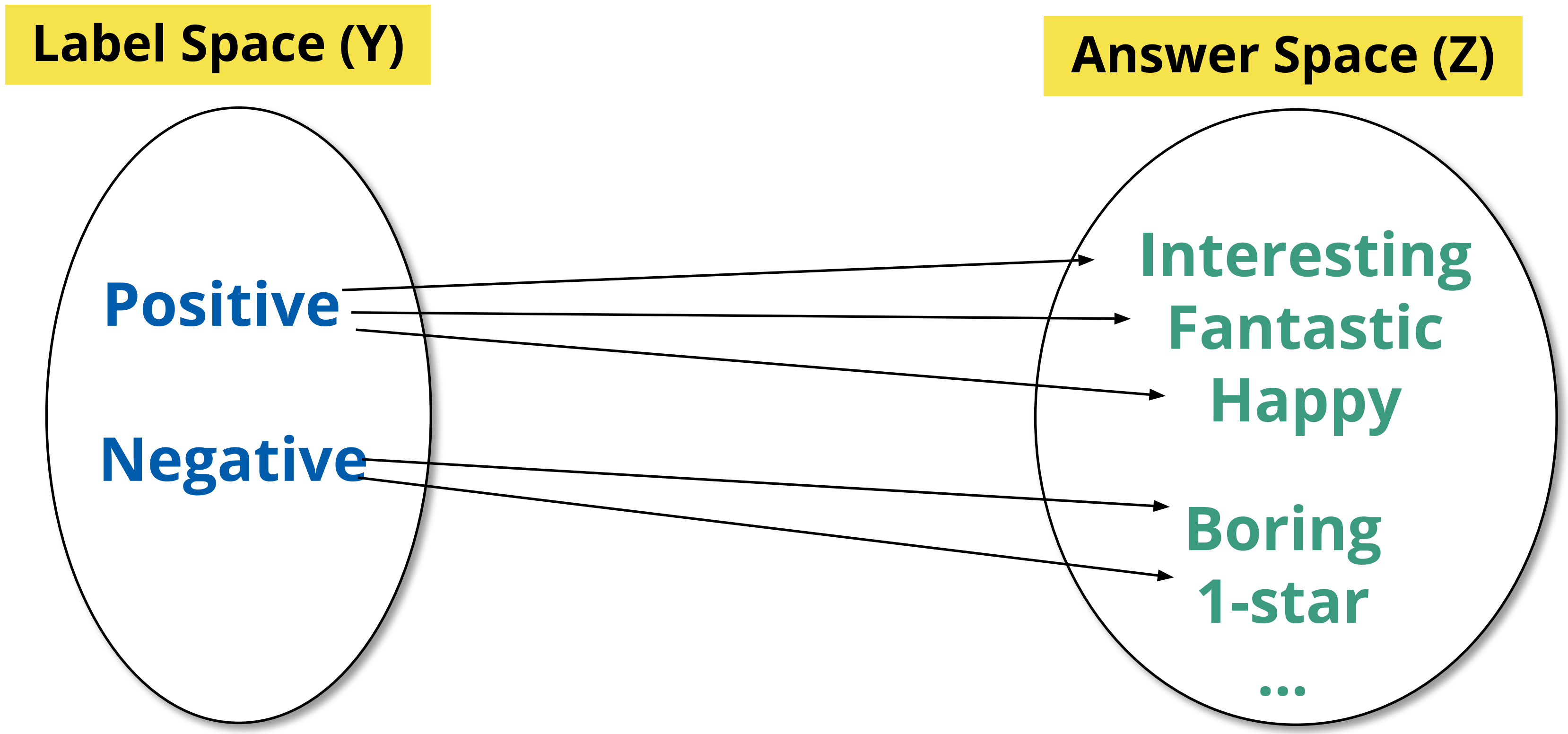
Research Question:

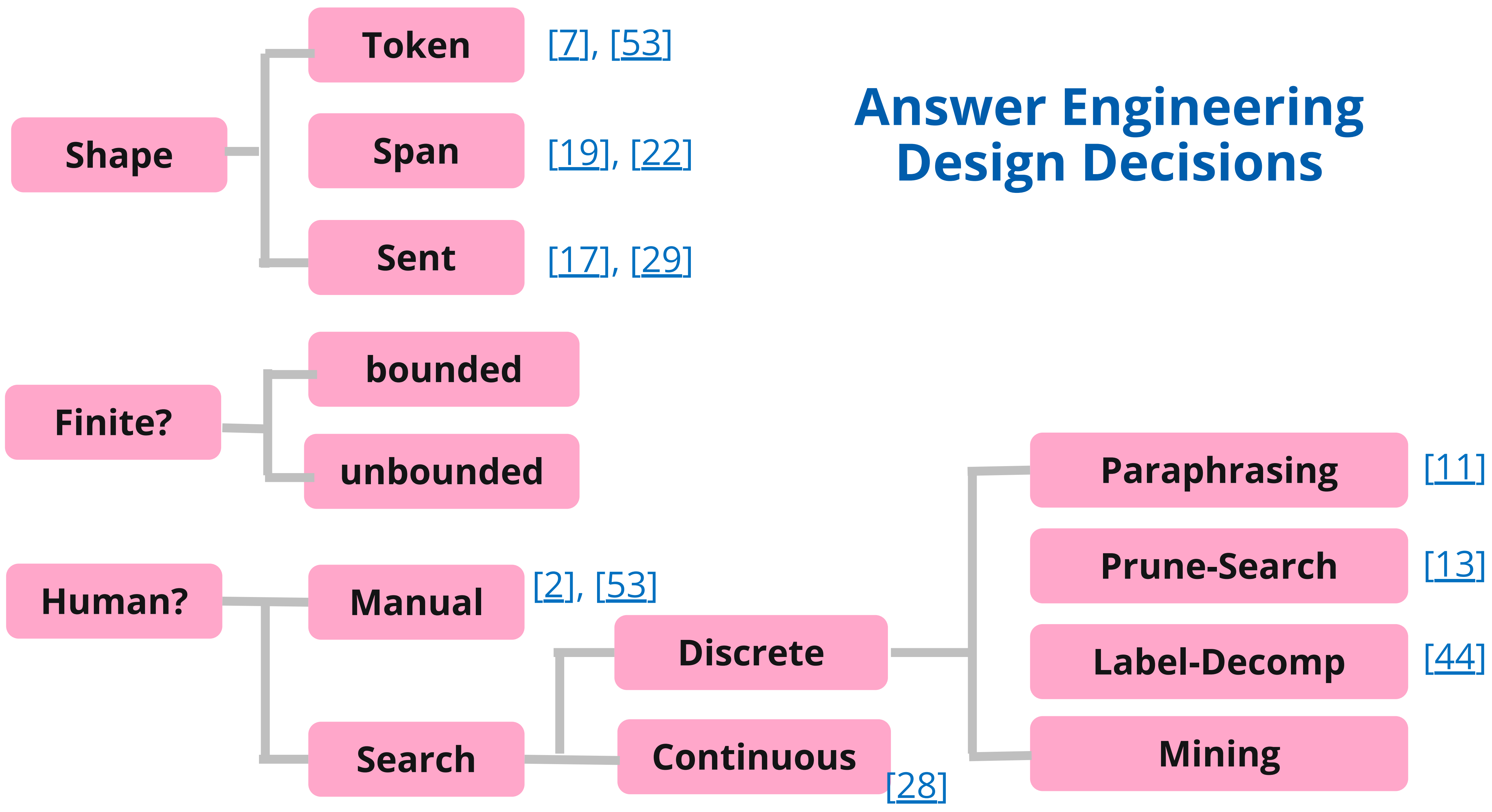
- Given a task (or a prompt), **how to define a suitable mapping function between label space and answer space?**

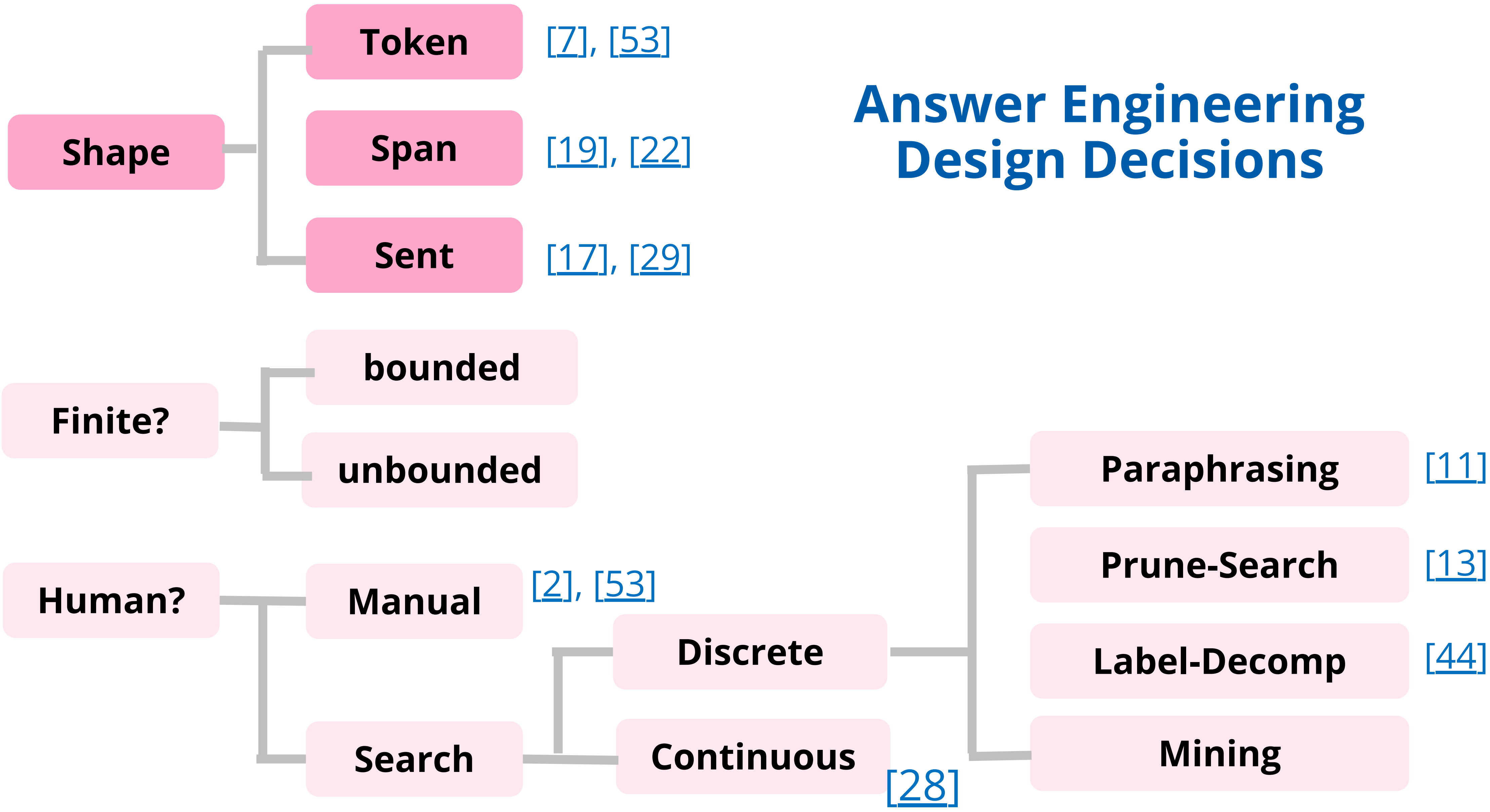
Answer Engineering

Research Question:

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




Answer Engineering
Design Decisions
Shape
Token
[7], [53]
Span
[19], [22]
Sent
[17], [29]
Finite?
bounded
unbounded
Human?
Manual
[2], [53]
Search
Discrete
Continuous
[28]
Paraphrasing
[11]
Prune-Search
[13]
Label-Decomp
[44]
Mining


Answer Engineering
Design Decisions
Shape
Token
[7], [53]
Span
[19], [22]
Sent
[17], [29]
Finite?
bounded
unbounded
Human?
Manual
[2], [53]
Search
Discrete
Continuous
[28]
Paraphrasing
[11]
Prune-Search
[13]
Label-Decomp
[44]
Mining

Answer Shape

Single Token

- Useful for most classification tasks
- Examples
  - <A movie review> The movie is fantastic/terrible.
  - <Premise> entails/contradicts <Hypothesis>

Answer Shape

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

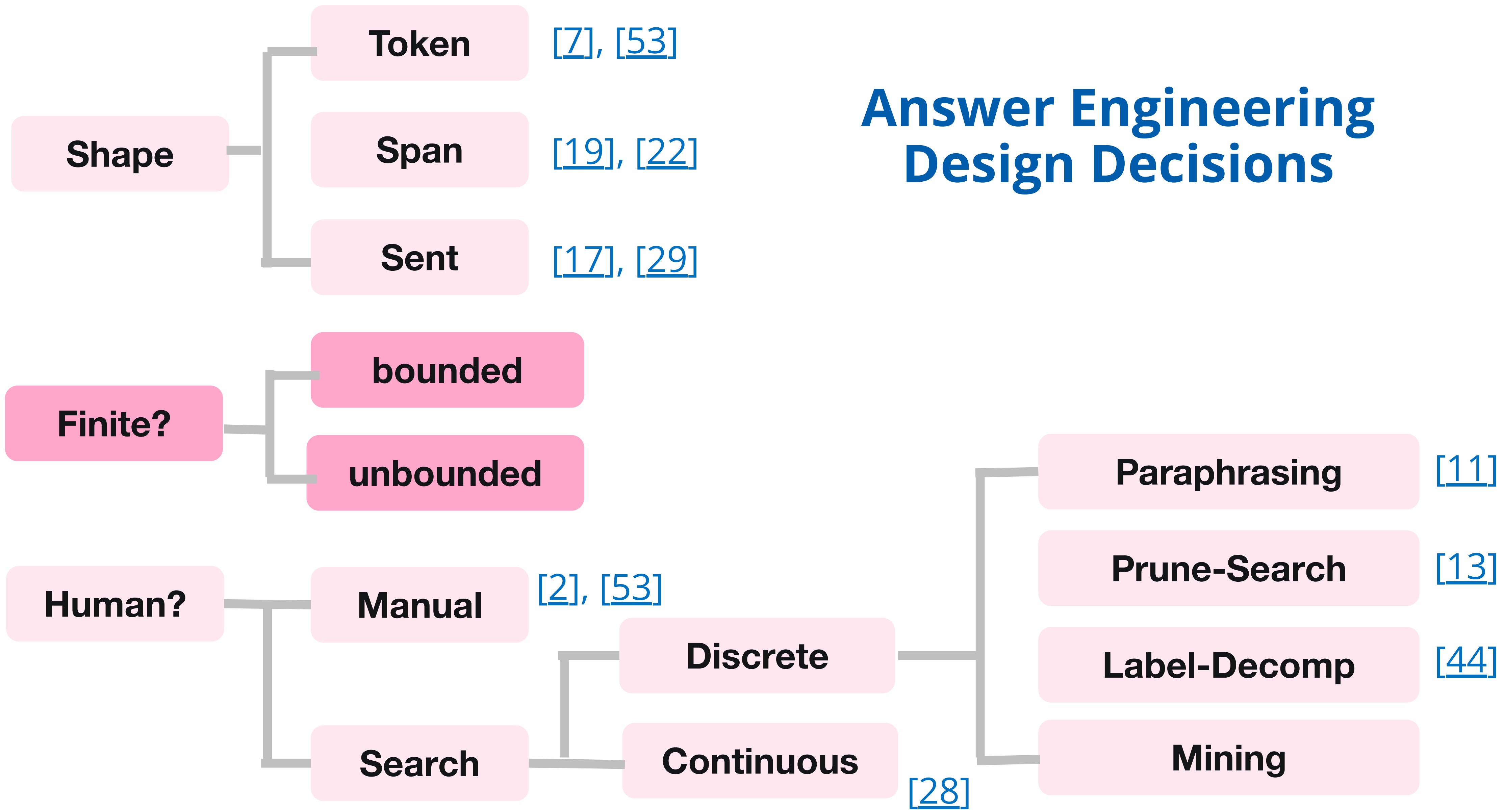
    (A) less gravity
    (B) more gravity
    (C) less friction [gold]
    (D) more friction

Answer Shape

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！


Answer Engineering Design Decisions
Shape
Token
[7], [53]
Span
[19], [22]
Sent
[17], [29]
Finite?
bounded
unbounded
Human?
Manual
[2], [53]
Search
Discrete
Continuous
[28]
Paraphrasing
[11]
Prune-Search
[13]
Label-Decomp
[44]
Mining

Answer Space

Bounded

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

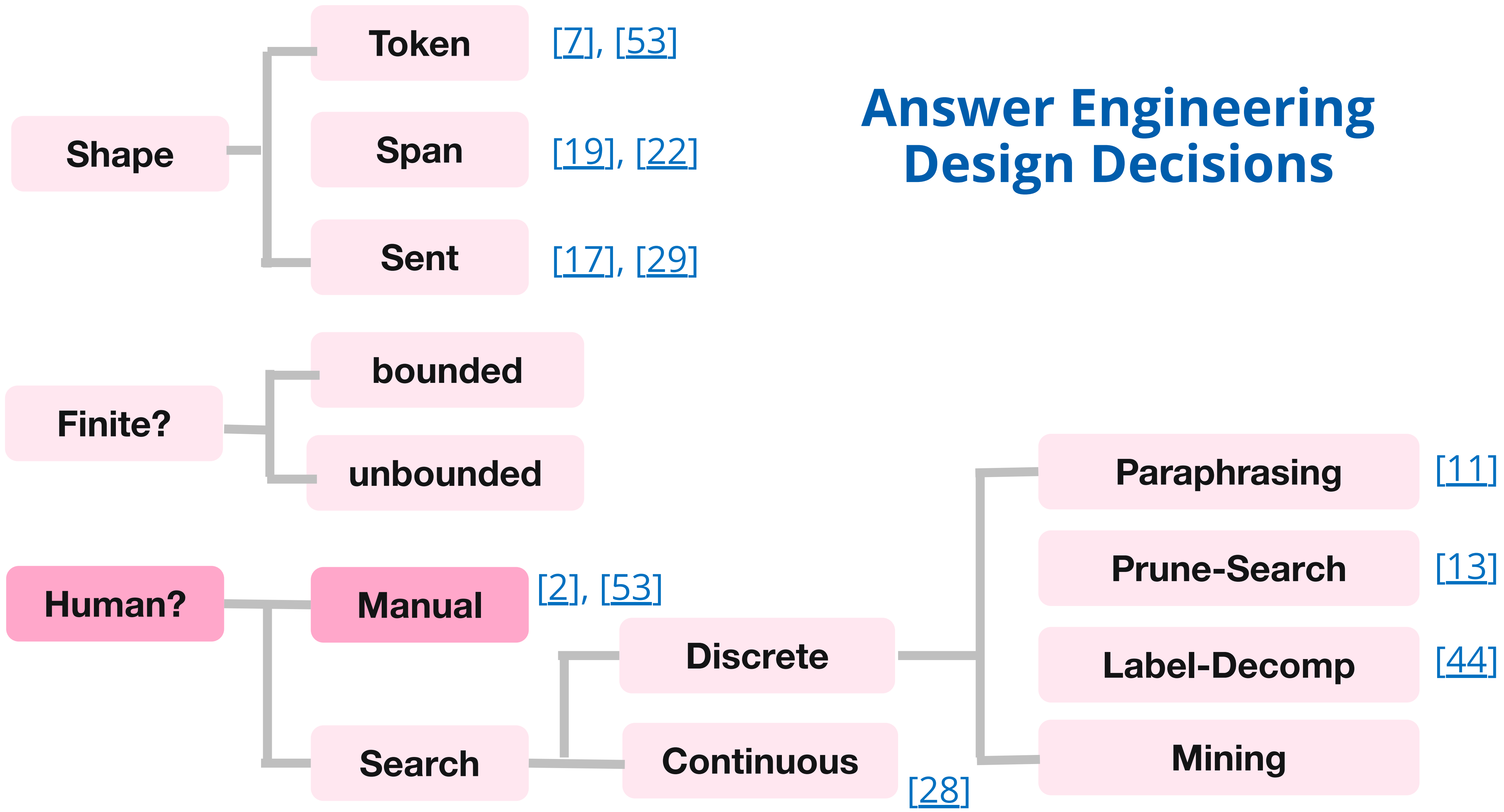
Answer Space

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.



Answer Engineering
Design Decisions
Shape
Token
[7], [53]
Span
[19], [22]
Sent
[17], [29]
Finite?
bounded
unbounded
Human?
Manual
[2], [53]
Search
Discrete
Continuous
[28]
Paraphrasing
[11]
Prune-Search
[13]
Label-Decomp
[44]
Mining

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

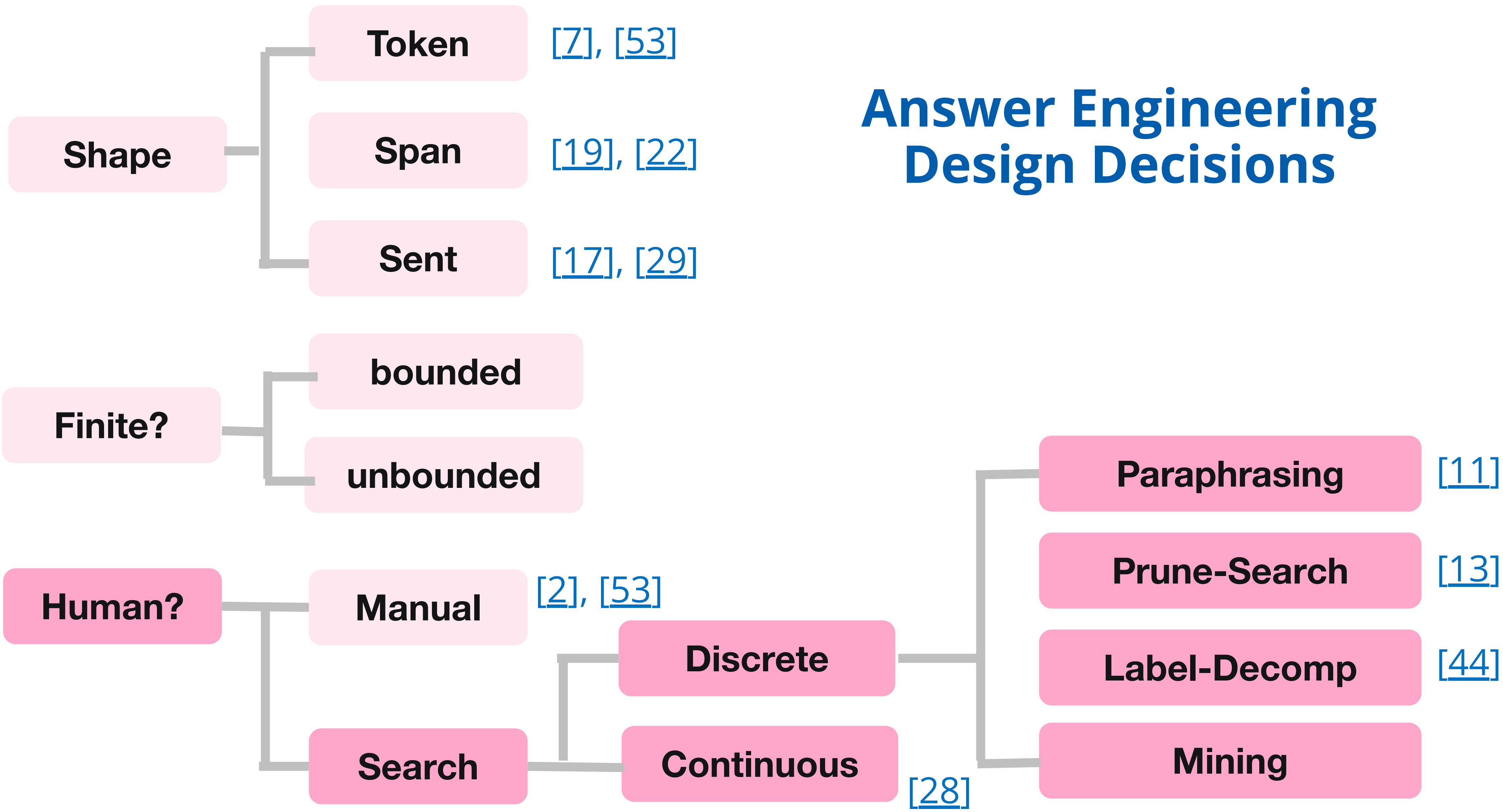
Human Design

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



Answer Engineering
Design Decisions
Shape
Token
[7], [53]
Span
[19], [22]
Sent
[17], [29]
Finite?
bounded
unbounded
Human?
Manual
[2], [53]
Search
Discrete
Continuous
[28]
Paraphrasing
[11]
Prune-Search
[13]
Label-Decomp
[44]
Mining

Design Decisions for Multiple Prompt Learning

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

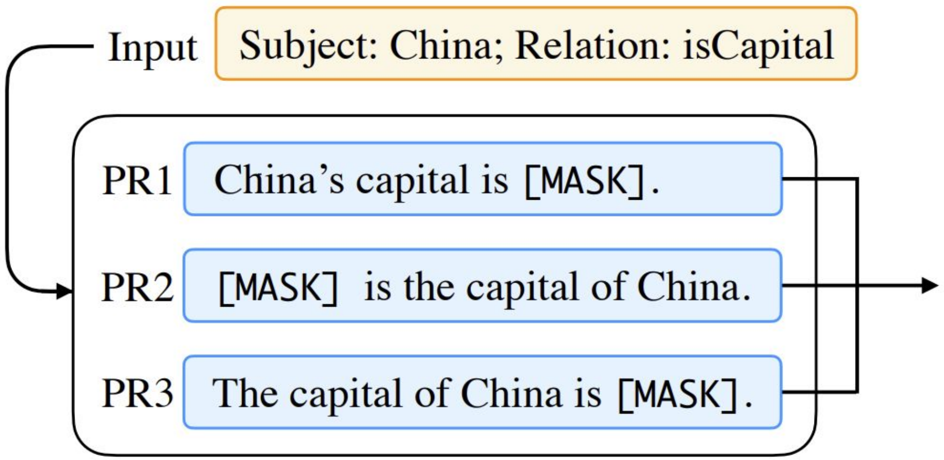
Prompt Ensembling

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

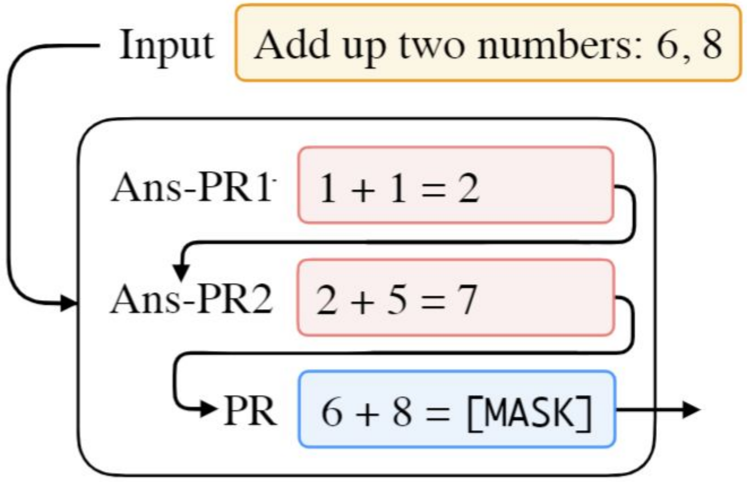
- Help the model answer the prompt with additional answered prompts

Advantage

- make use of the small amount of information that has been annotated

Core step

- Selection of answered prompts
- Ordering of answered prompts



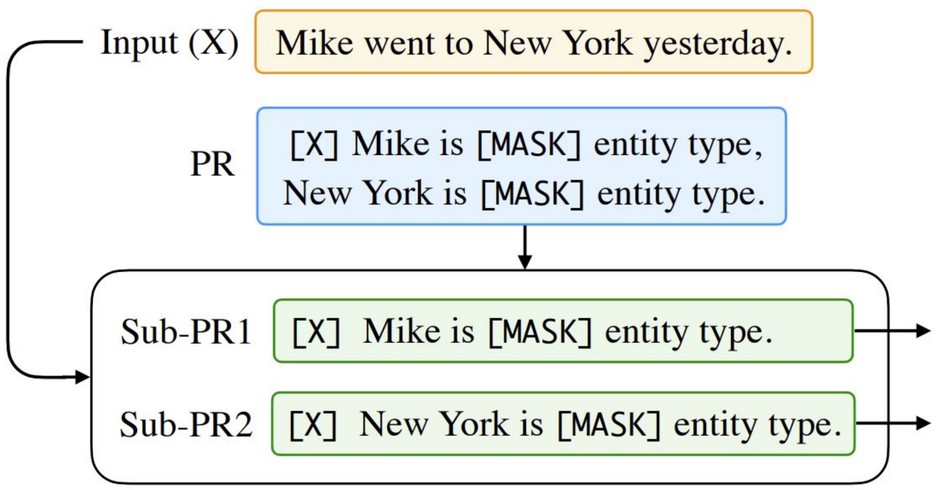
Prompt Decomposition

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.