中国科学院大学

University of Chinese Academy of Sciences

自然语言处理

第6讲 Prompt

王 石 资康莉 刘 瑜

2026年春季课程

https://ictkc.github.io/teaching/


国科大
UCAS
庚子
国科东路 南
UCAS East Road
SOUTH
北 国科东路
NORTH
UCAS East Road

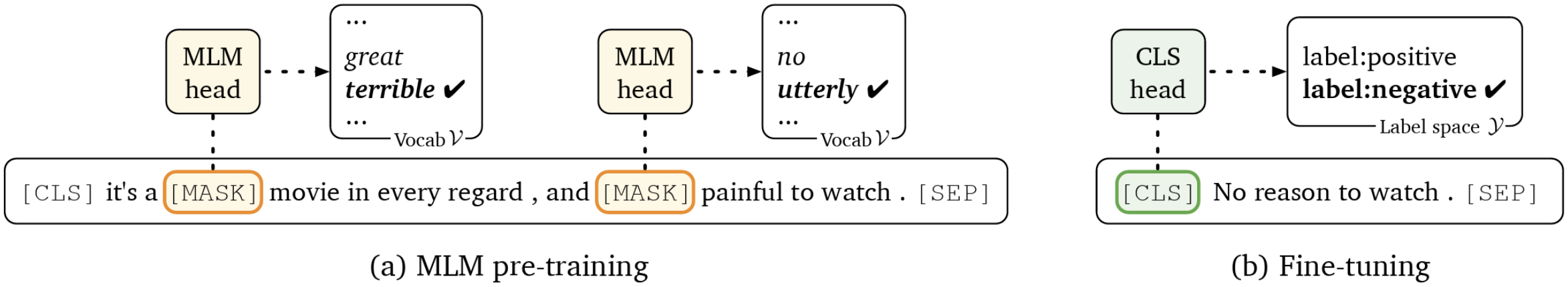
第6讲 Prompt

中国科学院大学
University of Chinese Academy of Sciences

目录

为什么需要提示?

- 传统预训练-微调范式的三大困境:
  - 任务目标差异大： 预训练目标与下游任务目标不一致



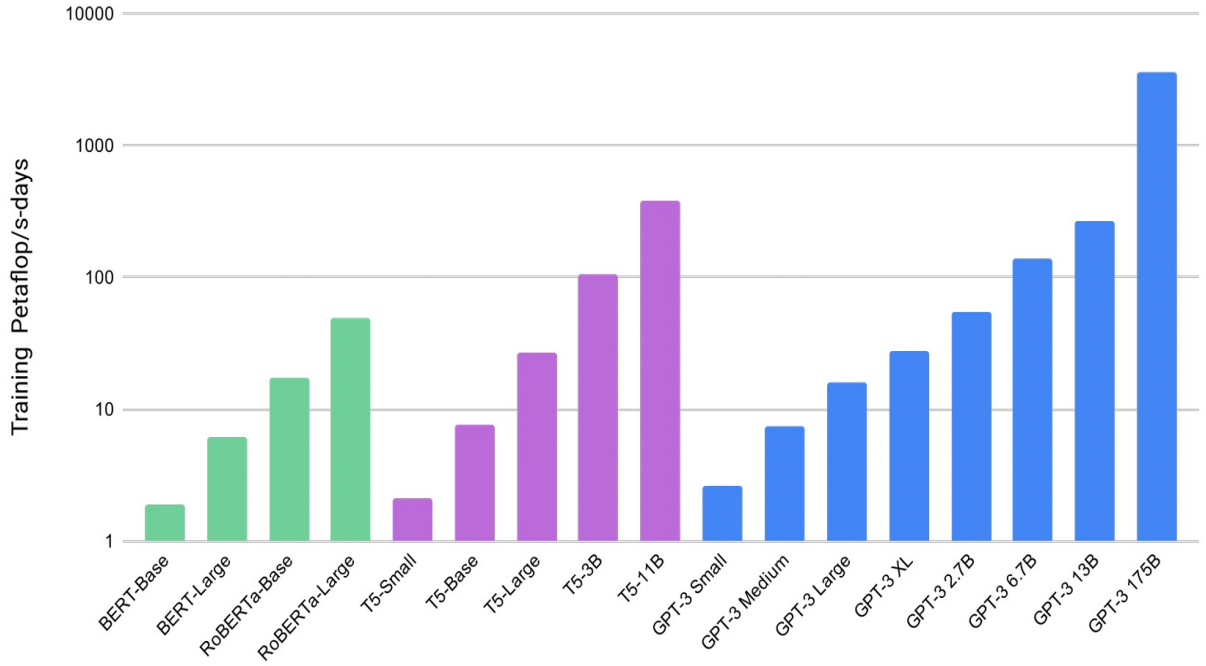
(a) MLM pre-training　　(b) Fine-tuning

Making Pre-trained Language Models Better Few-shot Learners. 2020.

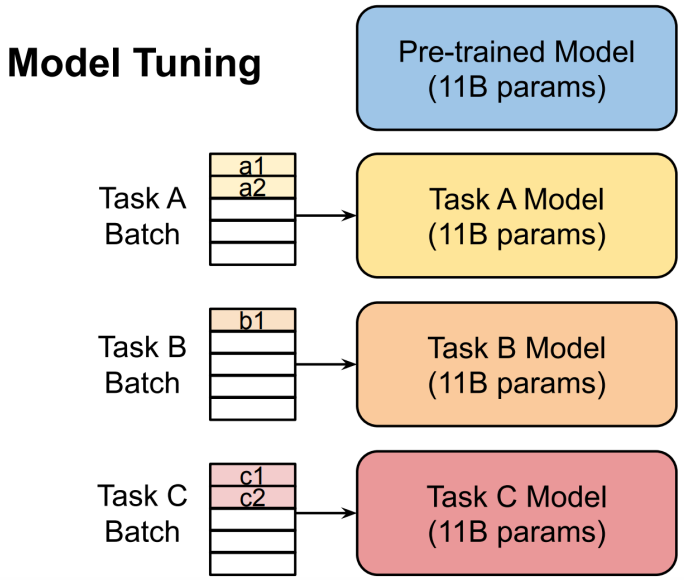
为什么需要提示?

- 传统预训练-微调范式的三大困境:
  - 微调成本高：语言模型越大，微调资源消耗也越高

模型参数规模逐渐增大



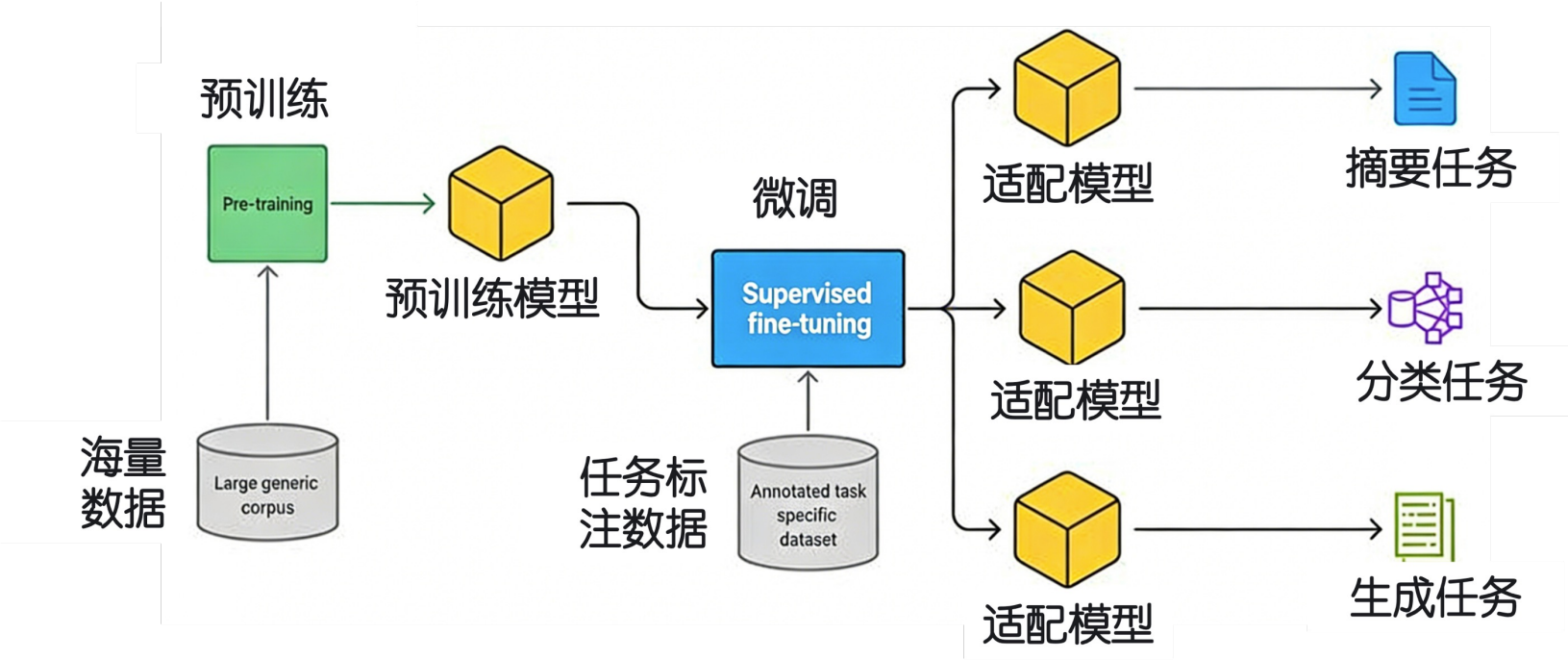
特定任务微调成本高



Language Models are Few-Shot Learners. 2021.

为什么需要提示?

- 传统预训练-微调范式的三大困境:
  - 泛化能力受限： 微调后的模型难以快速适应新任务或新领域



什么是提示?

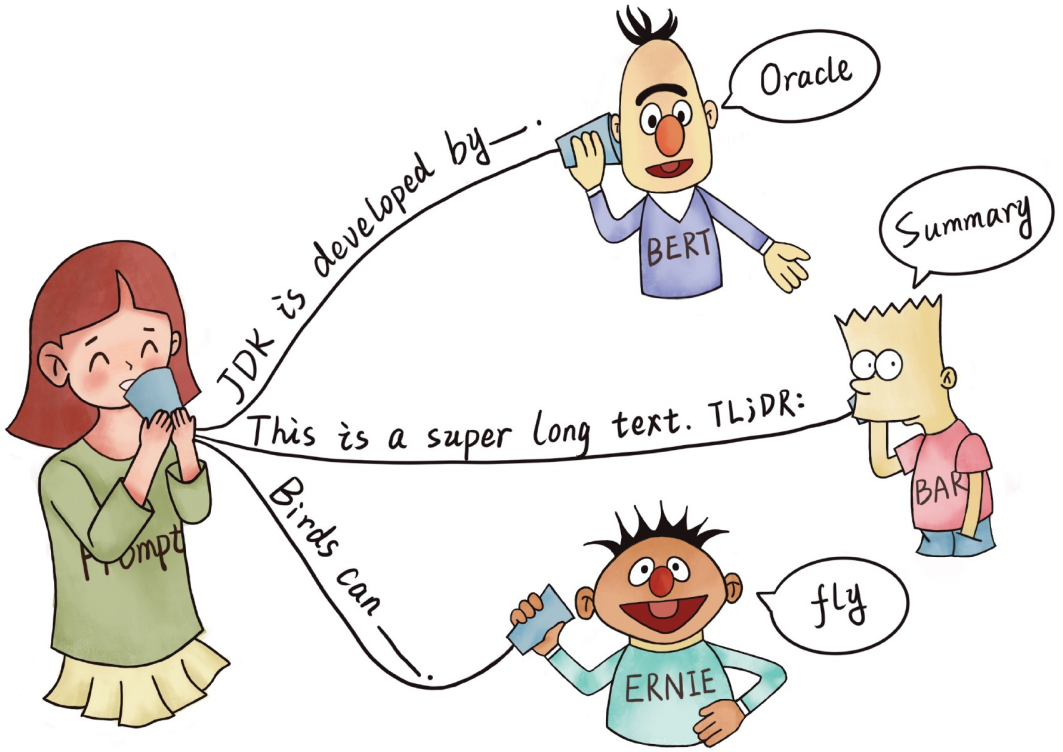
- 提示（Prompt）是输入给模型的一段信息，用来引导它生成符合预期的输出

- **示例**
  - 问题型：中国的首都是哪里?
  - 指令型：请把这句话翻译成英文：我爱自然语言处理
  - 示例型：Q: 2+2=? A: 4 \n Q: 3+3=? A:
  - 接口型：POST /api/translateText

    { “text”: “我爱自然语言处理”, "target_language": "en"}

什么是提示学习?

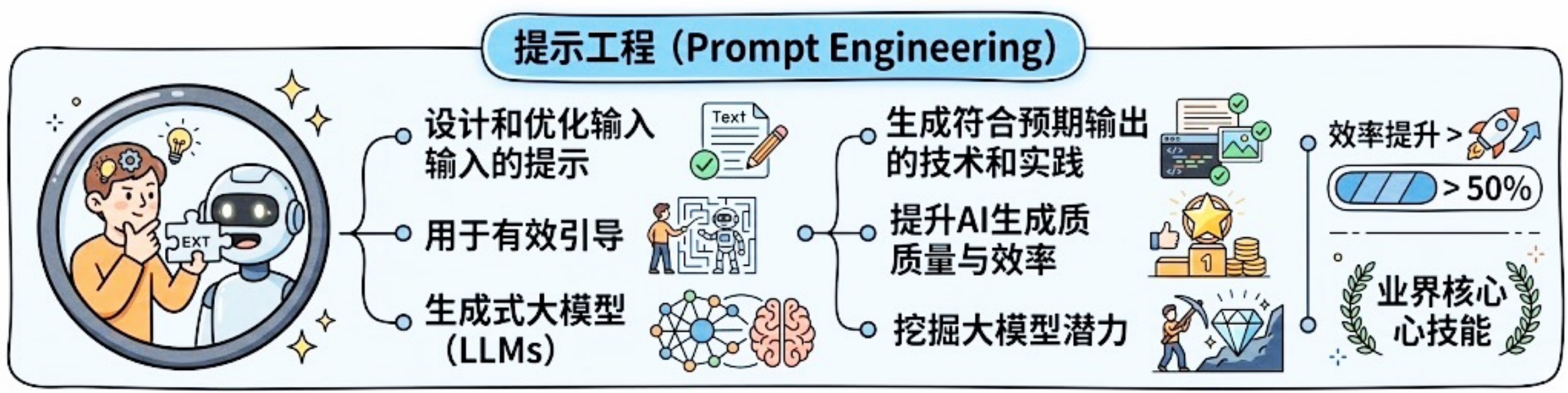
- 提示学习（Prompt Learning）通过**设计和学习提示**（prompt），将下游任务重新表述为**预训练模型**熟悉的形式



Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

什么是提示工程?

- 提示工程（Prompt Engineering）通过**精心设计输入提示**，以有效引导**大语言模型**生成符合预期输出的技术

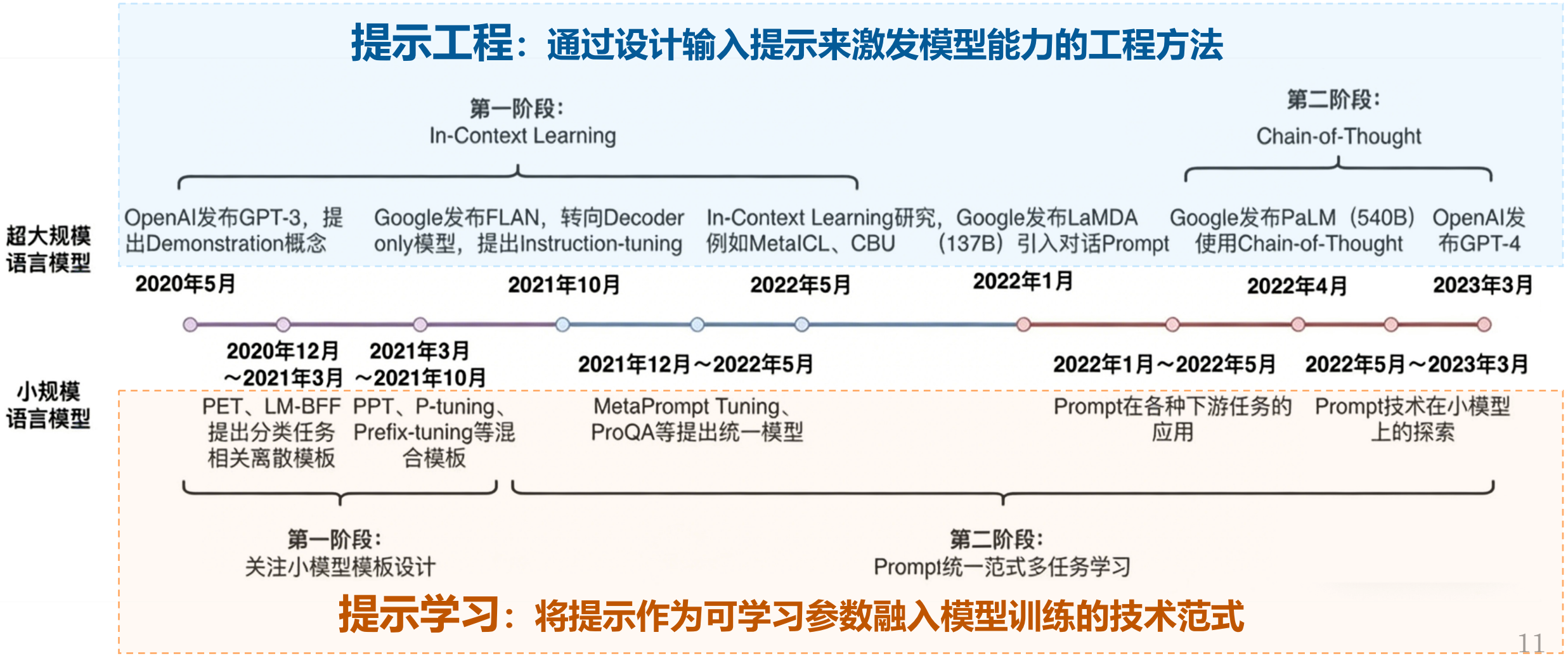


提示学习Vs提示工程

- 提示学习是“**学提示**”，提示工程是“**写提示**”

维度	提示学习	提示工程
优化方式	人工设计 / 自动学习	人工设计/ 自动学习
是否训练	是（优化prompt参数）	否（不改变参数）
技术门槛	较高（需训练）	低（编写提示）
灵活性	相对固定（需重新训练）	高（可随时更改）
代表方法	Prompt Tuning， Prefix Tuning	In-Context Learning，CoT

发展历程

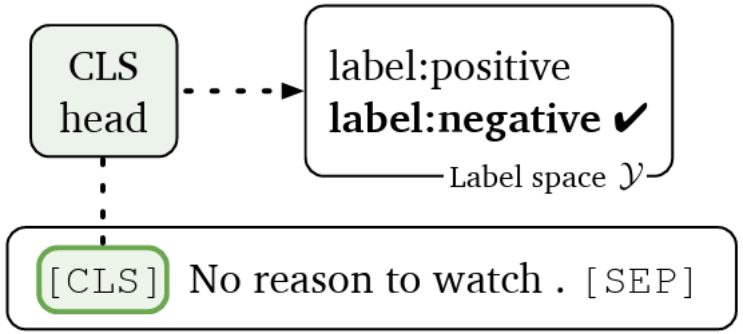

提示工程：通过设计输入提示来激发模型能力的工程方法
第一阶段：
In-Context Learning
第二阶段：
Chain-of-Thought
超大规模语言模型
OpenAI发布GPT-3，提出Demonstration概念
Google发布FLAN，转向Decoder only模型，提出Instruction-tuning
In-Context Learning研究，Google发布LaMDA例如MetaICL、CBU
Google发布PaLM（540B）（137B）引入对话Prompt
OpenAI发布GPT-4使用Chain-of-Thought
2020年5月
2021年10月
2022年5月
2022年1月
2022年4月
2023年3月
小规模语言模型
2020年12月～2021年3月
2021年3月～2021年10月
2021年12月～2022年5月
2022年1月～2022年5月
2022年5月～2023年3月
PET、LM-BFF提出分类任务相关离散模板
PPT、P-tuning、Prefix-tuning等混合模板
MetaPrompt Tuning、ProQA等提出统一模型
Prompt在各种下游任务的应用
Prompt技术在小模型上的探索
第一阶段：
关注小模型模板设计
第二阶段：
Prompt统一范式多任务学习
提示学习：将提示作为可学习参数融入模型训练的技术范式

中国科学院大学
University of Chinese Academy of Sciences

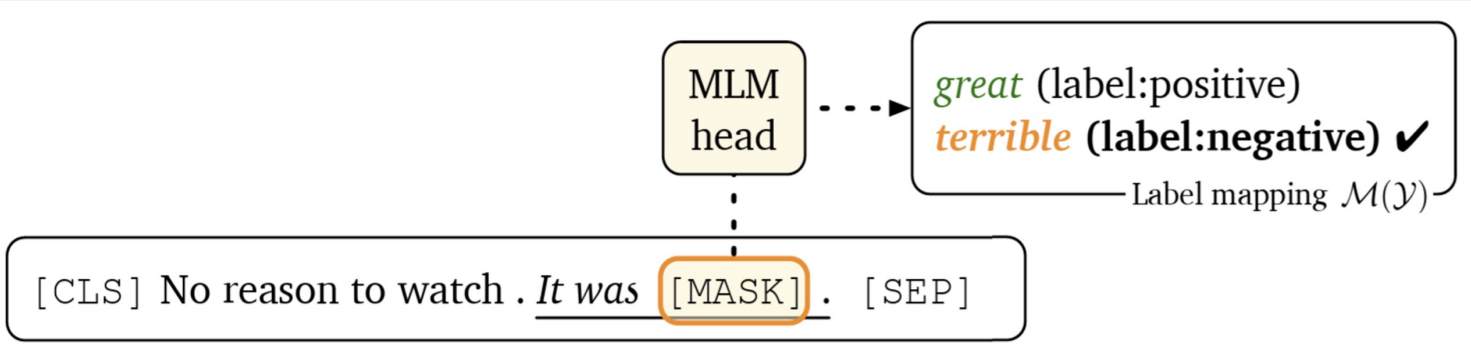
目 录

提示学习

- 提示学习（Prompt Learning）通过设计并学习提示（prompt），引导模型在**不（或少量）更新参数**的情况下完成下游任务
- 新范式：“预训练+提示学习”

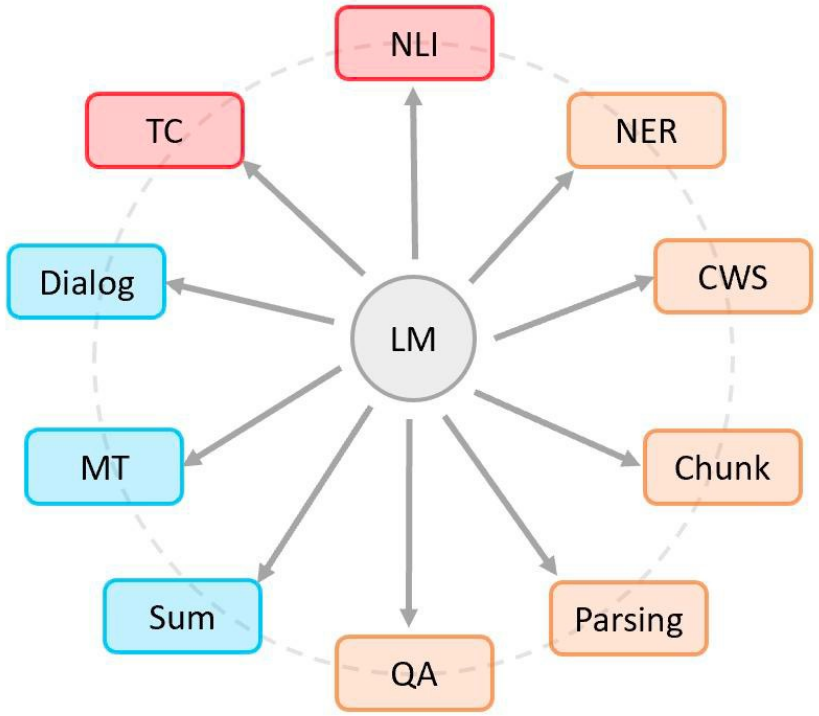


基于Fine-tune的情感分析任务

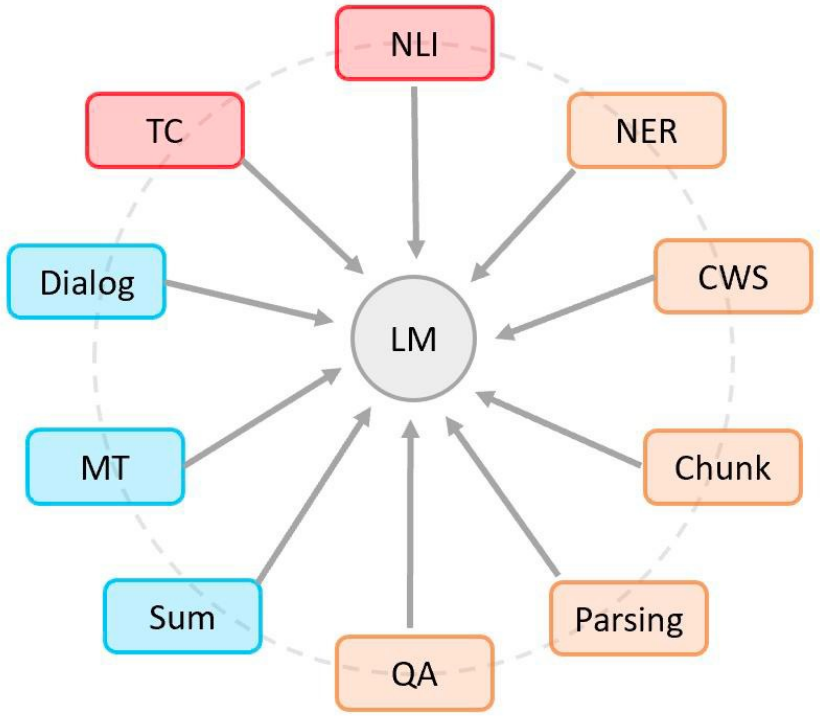


基于Prompt的情感分析任务

微调范式Vs提示范式



预训练+微调范式



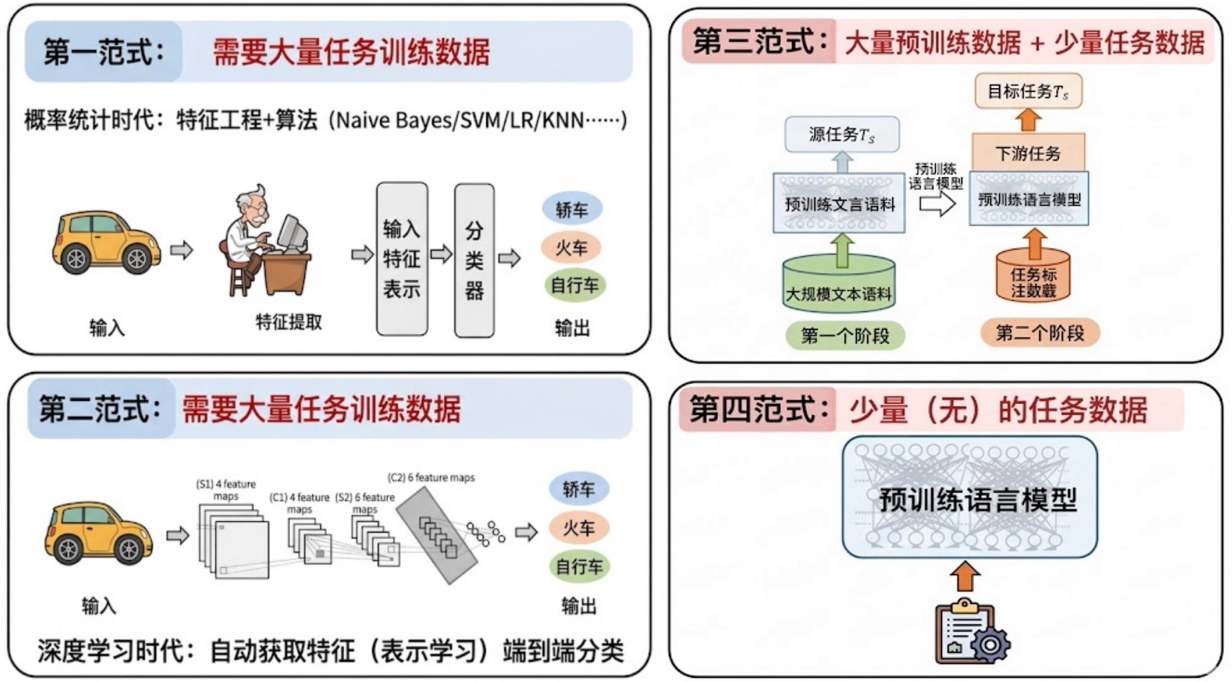
预训练+提示范式

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

不同范式比较

❑ NLP领域发展四个范式：

- **第一范式**：概率统计时代的完全监督学习，人工设计和定义特征模板
- **第二范式**：深度学习时代的完全监督学习，探究适配下游任务的结构偏置
- **第三范式**：预训练+微调范式，引入额外的目标函数到预训练语言模型，以便让其更适配下游任务，
- **第四范式**：预训练+提示范式，通过设计合适的prompt实现对下游任务建模方式的重新定义



Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

提示学习的定义

- 监督学习
  - 给定输入x，预测输出y，监督学习模型： $P(\boldsymbol{y}|\boldsymbol{x};\theta)$
  - 情感分析示例： ***Input x***: *I love this movie.*

    Output y: ++(very positive) Y={++, +, ~, -, --}

- 提示学习
  - 从学习$P(\boldsymbol{y}|\boldsymbol{x};\theta)$改为学习$P(\boldsymbol{x};\theta)$，再去预测$y$
  - 通过引入模板将输入x调整为完形填空格式的 $\boldsymbol{x}' = f_{\text{prompt}}(\boldsymbol{x})$，调整后的输入中包含一些空槽，利用语言模型 预测槽值进而推断出y

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

提示学习的定义

- 提示模板f_{prompt}
  - 采用模板转换，包括一个输入槽[X]和一个答案槽[Z]，将输入x填入槽[X]

Source Input x: *I love this movie.*

Template: *[X] Overall, it was a [Z] movie*

Prompt Input x': *I love this movie. Overall, it was a [Z] movie*

  - 模板可以是自然语言token或非自然语言token
  - 答案[Z]可在句中（cloze prompt，NLU常用)或句末（prefix prompt，NLG常用），槽的数量可以为任意个

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

提示学习的定义

❑ 答案搜索

- 将x'输入到语言模型，从Z中搜索使得语言模型得分最高的候选槽值

$$\hat{z} = \underset{z \in Z}{search} P(f_{\text{fill}}(x', z); \theta)$$

- Z可以包括词表中所有的token（生成任务），也可以是一个特定标签集合（如分类任务）
- 示例：　Z={“excellent”，“good”，“OK”，“bad”，“horrible”}
  Y={++, +, ~, -, --}

❑ 答案映射

- 将得到的答案Z与对应任务的标签Y做1-1或N -1映射

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

Prompt术语形式化表示

Name	Notation	Example	Description
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	`[X]` Overall, it was a `[Z]` movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot `[Z]` where answer $\boldsymbol{z}$ may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a `[Z]` movie.	A text where `[X]` is instantiated by input $\boldsymbol{x}$ but answer slot `[Z]` is not.
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot `[Z]` is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot `[Z]` is filled with a true answer.
Answer	$\boldsymbol{z}$	"good", "fantastic", "boring"	A token, phrase, or sentence that fills `[Z]`

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.

不同任务的Prompt

Type	Task	Input ([X])	Template	Answer ([Z])
Text CLS	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1][X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...

提示学习的优势

❑ 提示学习的四个核心优势：

降低对标注数据的依赖

能够在小样本（Few-shot）甚至零样本（Zero -shot）场景下工作，极大地减少了对标注数据的需求。

提升模型泛化能力

通过利用预训练模型的通用知识，模型能够更好地泛化到新任务和新领域，适应不同的应用场景。

参数高效

无需微调模型的全部参数，仅需调整少量提示相关的参数（如Prompt Tuning），大大降低了计算成本。

增强可解释性

提示模板的设计使得模型的决策过程更加透明，部分解决了黑箱问题，让推理逻辑更易于理解。

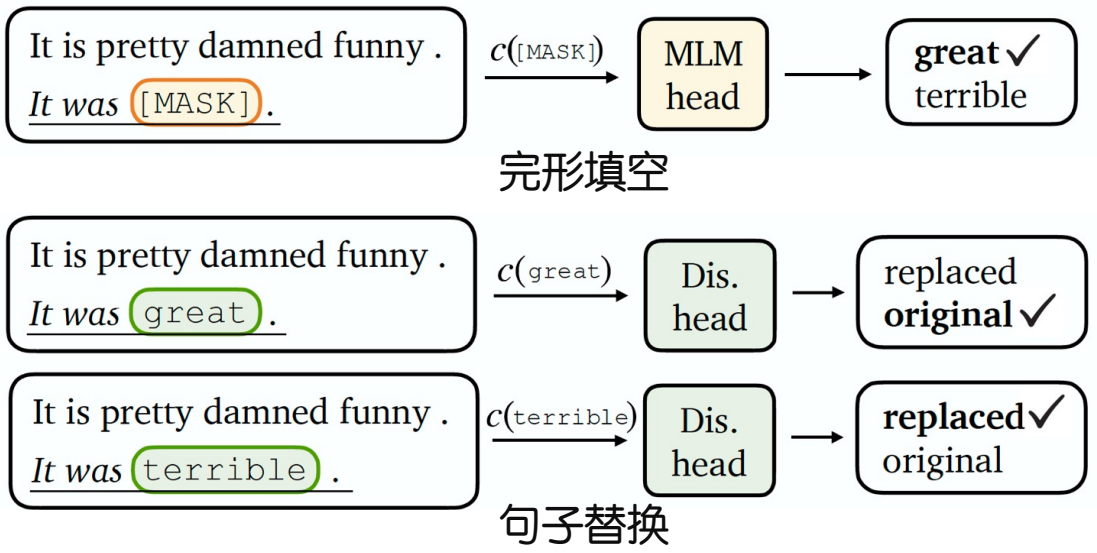
提示学习的局限性

- 提示学习面临的两个问题
  - 人类认为不错的提示对于LM来说不一定是一个好的提示，这个性质被称为提示的sub-optimal（次优）性
  - 提示的选择对于预训练模型的影响非常大

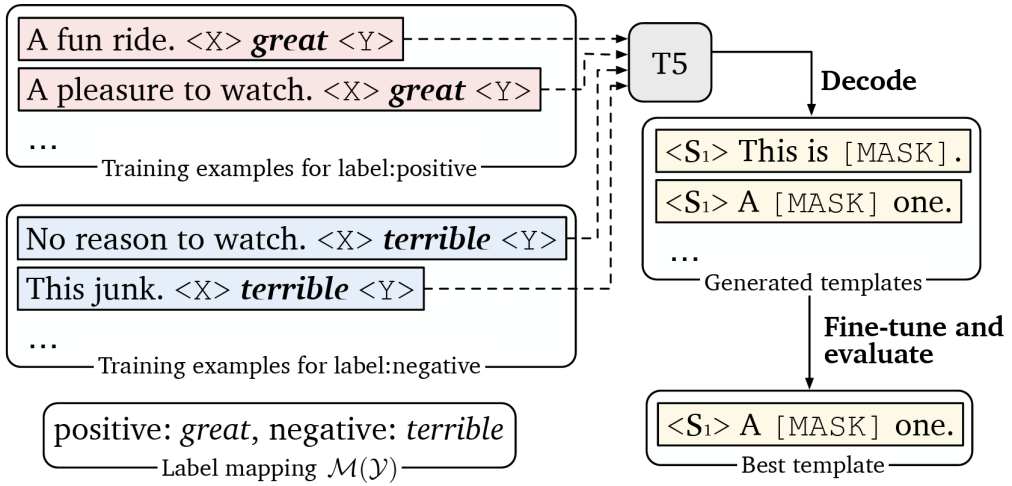
Prompt	P@1
[X] is located in [Y]. *(original)*	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

提示学习的分类

- 硬提示/离散提示
  - 使用自然语言手工设计的提示模板，引导模型完成任务



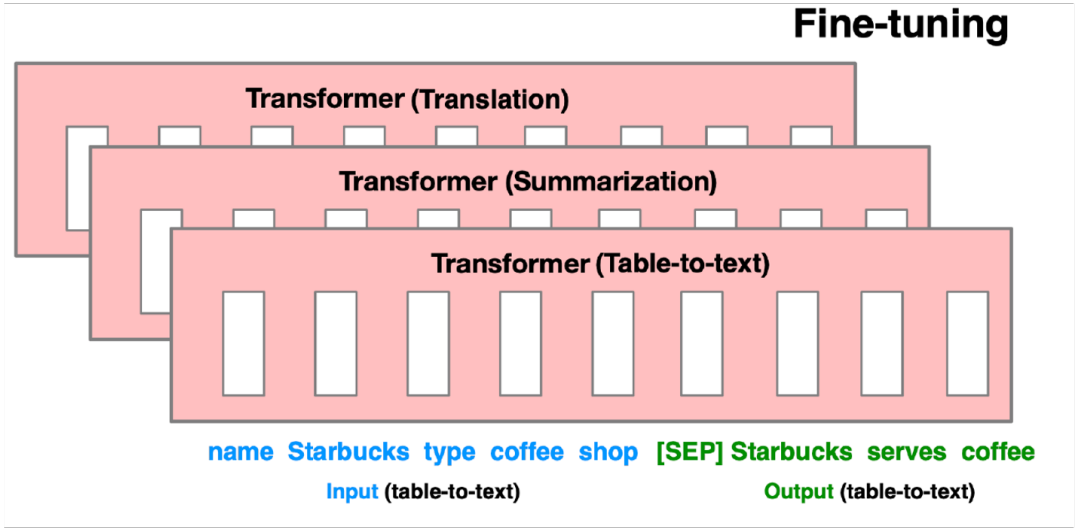
人工编写提示模版



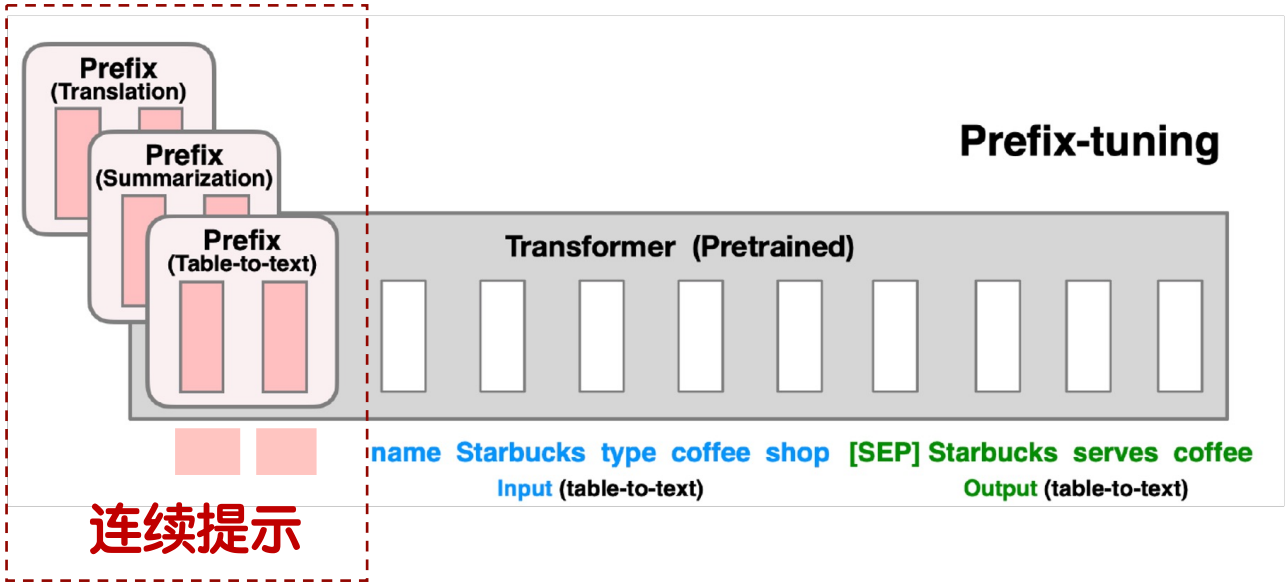
自动生成提示模版

提示学习的分类

- 软提示/连续提示
  - 将提示表示为可学习的连续向量，通过训练自动优化

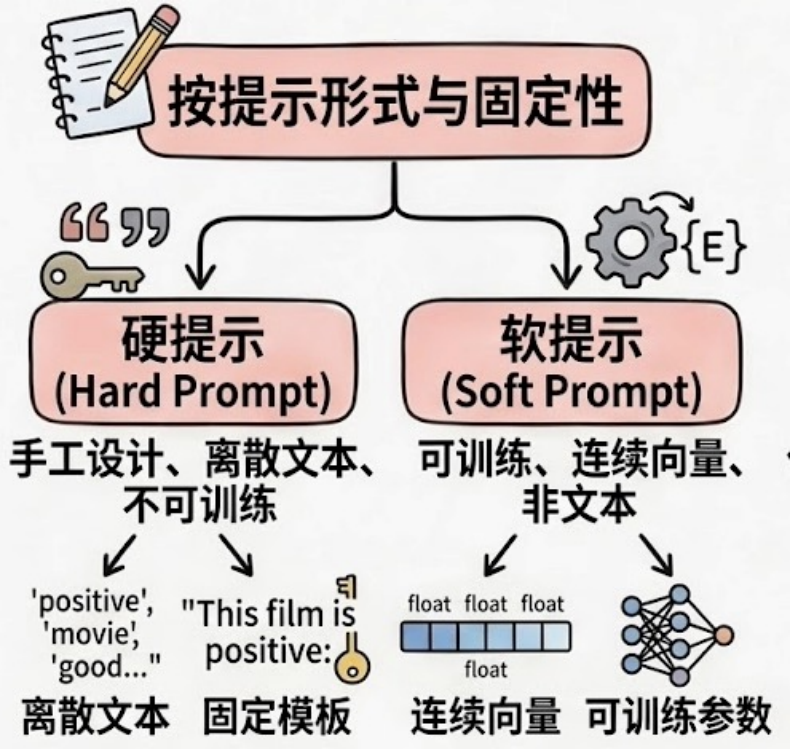


传统微调策略



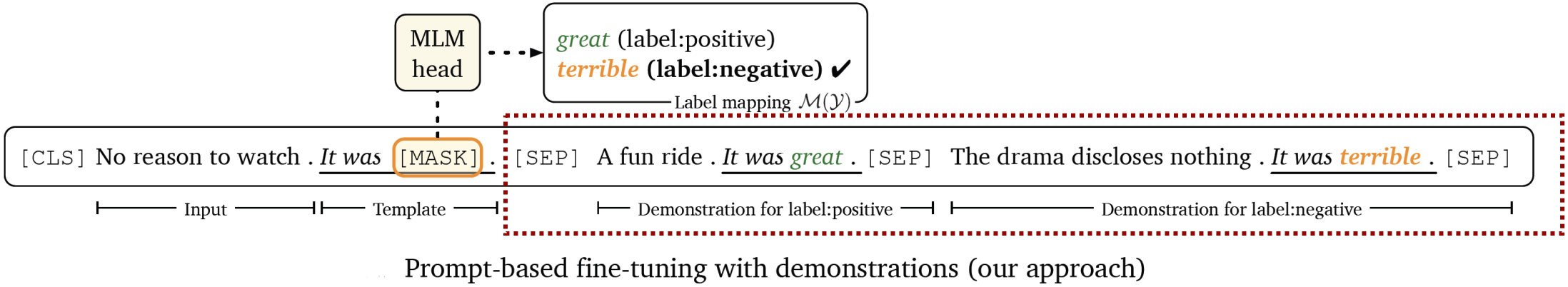
提示学习技术

- 硬提示方法
  - LM-BFF
  - BET
  - AutoPrompt
- 软提示方法
  - Prefix tuning
  - Prompt Tuning
  - P-tuning / v2



LM-BFF

- 方法介绍
  - 使用小规模LM模型（如BERT/Roberta）在Few-shot上进行实验
  - 受GPT-3 ICL的启发，拼接部分样例到当前输入后，每个类别只选取一个语义最相似的样例：



Prompt-based fine-tuning with demonstrations (our approach)

Making Pre-trained Language Models Better Few-shot Learners. 2021.(陈丹琦, 普林斯顿大学).

LM-BFF

❑ 模板生成

- 手工设计的模版对模型性能影响的比较大
- 用T5模型生成<X>与<Y>对应的span，从而生成候选模板：

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

Table 2: The impact of templates and label words on prompt-based fine-tuning ($K = 16$).

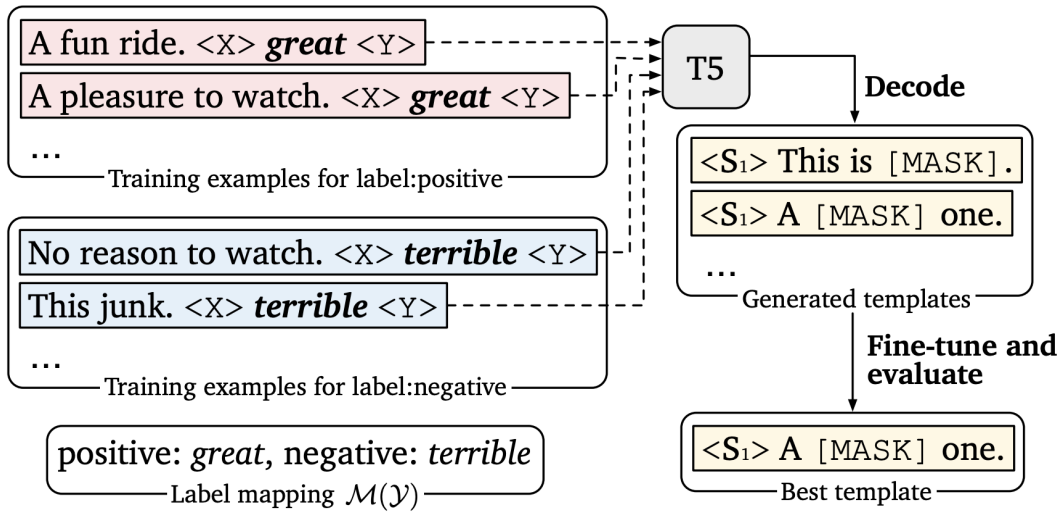


Figure 2: Our approach for template generation.

LM-BFF

❑ 实验结果

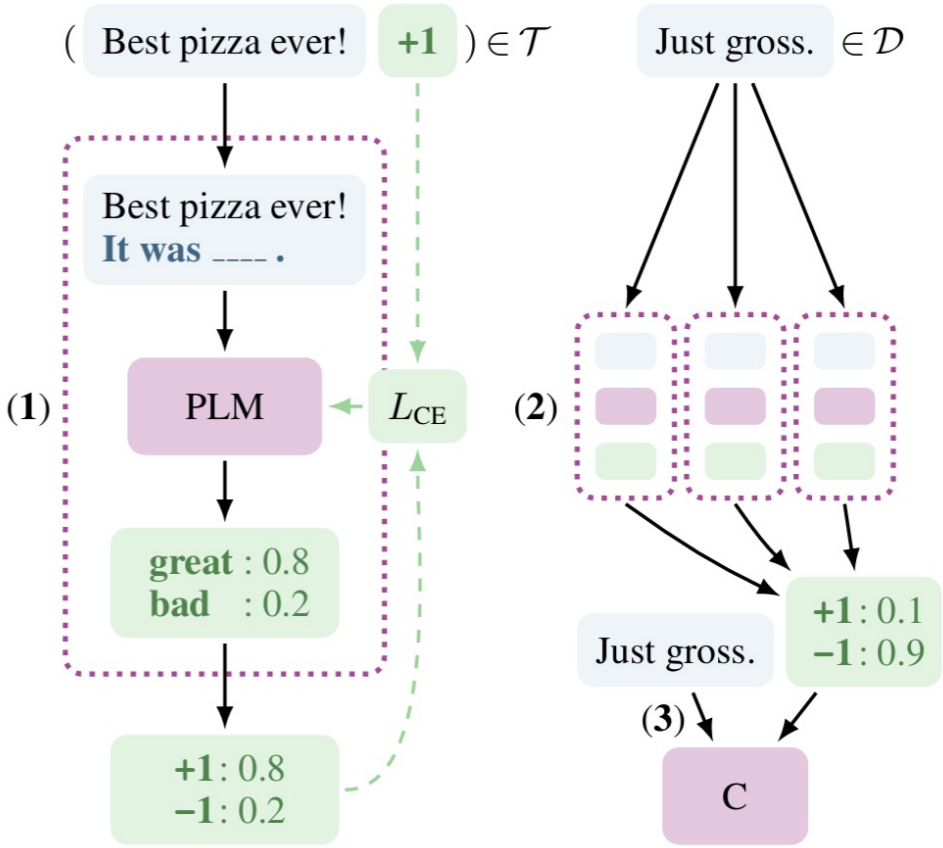
	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (Matt.)
Majority†	*50.9*	*23.1*	*50.0*	*50.0*	*50.0*	*50.0*	*18.8*	*0.0*
Prompt-based zero-shot‡	83.6	35.0	80.8	79.5	67.6	51.4	32.0	2.0
"GPT-3" in-context learning	84.8 (1.3)	30.6 (0.9)	80.5 (1.7)	87.4 (0.8)	63.8 (2.1)	53.6 (1.0)	26.2 (2.4)	-1.5 (2.4)
Fine-tuning	81.4 (3.8)	43.9 (2.0)	76.9 (5.9)	75.8 (3.2)	72.0 (3.8)	90.8 (1.8)	88.8 (2.1)	**33.9** (14.3)
Prompt-based FT (man)	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	84.7 (2.2)	91.2 (1.1)	84.8 (5.1)	9.3 (7.3)
+ demonstrations	92.6 (0.5)	**50.6** (1.4)	86.6 (2.2)	90.2 (1.2)	**87.0** (1.1)	**92.3** (0.8)	87.5 (3.2)	18.7 (8.8)
Prompt-based FT (auto)	92.3 (1.0)	49.2 (1.6)	85.5 (2.8)	89.0 (1.4)	85.8 (1.9)	91.2 (1.1)	88.2 (2.0)	14.0 (14.1)
+ demonstrations	**93.0** (0.6)	49.5 (1.7)	**87.7** (1.4)	**91.0** (0.9)	86.5 (2.6)	91.4 (1.8)	**89.4** (1.7)	21.8 (15.9)
Fine-tuning (full)†	*95.0*	*58.7*	*90.8*	*89.4*	*87.8*	*97.0*	*97.4*	*62.6*

	MNLI (acc)	MNLI-mm (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	STS-B (Pear.)
Majority†	*32.7*	*33.0*	*33.8*	*49.5*	*52.7*	*81.2*	*0.0*	-
Prompt-based zero-shot‡	50.8	51.7	49.5	50.8	51.3	61.9	49.7	-3.2
"GPT-3" in-context learning	52.0 (0.7)	53.4 (0.6)	47.1 (0.6)	53.8 (0.4)	60.4 (1.4)	45.7 (6.0)	36.1 (5.2)	14.3 (2.8)
Fine-tuning	45.8 (6.4)	47.8 (6.8)	48.4 (4.8)	60.2 (6.5)	54.4 (3.9)	76.6 (2.5)	60.7 (4.3)	53.5 (8.5)
Prompt-based FT (man)	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)	71.0 (7.0)
+ demonstrations	**70.7** (1.3)	**72.0** (1.2)	**79.7** (1.5)	**69.2** (1.9)	68.7 (2.3)	77.8 (2.0)	**69.8** (1.8)	73.5 (5.1)
Prompt-based FT (auto)	68.3 (2.5)	70.1 (2.6)	77.1 (2.1)	68.3 (7.4)	**73.9** (2.2)	76.2 (2.3)	67.0 (3.0)	75.0 (3.3)
+ demonstrations	70.0 (3.6)	**72.0** (3.1)	77.5 (3.5)	68.5 (5.4)	71.1 (5.3)	**78.1** (3.4)	67.7 (5.8)	**76.4** (6.2)
Fine-tuning (full)†	*89.8*	*89.5*	*92.6*	*93.3*	*80.9*	*91.4*	*81.7*	*91.9*

PET

❑ 方法介绍

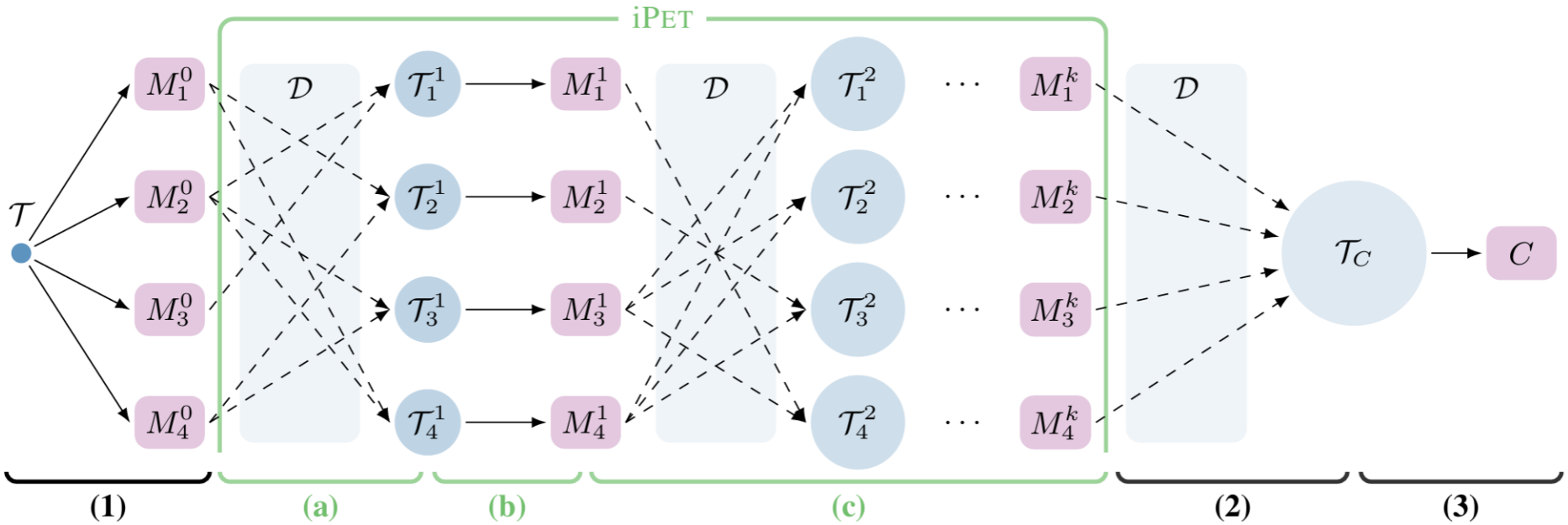
- 构造PVP： Pattern-Verbalizer Pair，将任务改写为MLM形式
- 微调模型：少样本数据微调LM模型
- 多PVP集成：设计不同的模版，构建多个prompt，多个模型的结果投票
- 半监督方法：利用少量标签数据+大量无标签数据 （软标签）



Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. 2021.

PET

- 增强版本：**iPET（iterative PET）** 是 PET 的增强版本，采用多轮迭代训练，每一轮利用更强的模型生成更高质量的伪标签



PET

❑ 实验结果

在极少数据下，PET/iPET 的效果明显优于传统监督学习

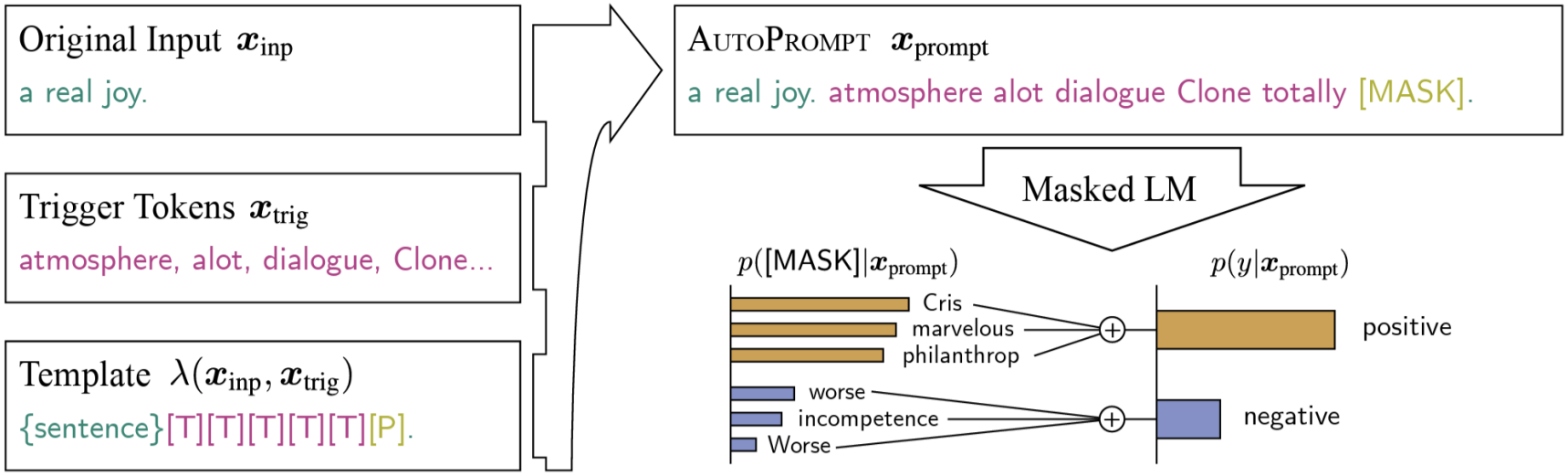
Line	Examples	Method	Yelp	AG's	Yahoo	MNLI (m/mm)
1	$\|\mathcal{T}\| = 0$	unsupervised (avg)	33.8 ±9.6	69.5 ±7.2	44.0 ±9.1	39.1 ±4.3 / 39.8 ±5.1
2		unsupervised (max)	40.8 ±0.0	79.4 ±0.0	56.4 ±0.0	43.8 ±0.0 / 45.0 ±0.0
3		iPET	**56.7** ±0.2	**87.5** ±0.1	**70.7** ±0.1	**53.6** ±0.1 / **54.2** ±0.1
4	$\|\mathcal{T}\| = 10$	supervised	21.1 ±1.6	25.0 ±0.1	10.1 ±0.1	34.2 ±2.1 / 34.1 ±2.0
5		PET	52.9 ±0.1	87.5 ±0.0	63.8 ±0.2	41.8 ±0.1 / 41.5 ±0.2
6		iPET	**57.6** ±0.0	**89.3** ±0.1	**70.7** ±0.1	**43.2** ±0.0 / **45.7** ±0.1
7	$\|\mathcal{T}\| = 50$	supervised	44.8 ±2.7	82.1 ±2.5	52.5 ±3.1	45.6 ±1.8 / 47.6 ±2.4
8		PET	60.0 ±0.1	86.3 ±0.0	66.2 ±0.1	63.9 ±0.0 / 64.2 ±0.0
9		iPET	**60.7** ±0.1	**88.4** ±0.1	**69.7** ±0.0	**67.4** ±0.3 / **68.3** ±0.3
10	$\|\mathcal{T}\| = 100$	supervised	53.0 ±3.1	86.0 ±0.7	62.9 ±0.9	47.9 ±2.8 / 51.2 ±2.6
11		PET	61.9 ±0.0	88.3 ±0.1	69.2 ±0.0	74.7 ±0.3 / 75.9 ±0.4
12		iPET	**62.9** ±0.0	**89.6** ±0.1	**71.2** ±0.1	**78.4** ±0.7 / **78.6** ±0.5
13	$\|\mathcal{T}\| = 1000$	supervised	63.0 ±0.5	**86.9** ±0.4	70.5 ±0.3	73.1 ±0.2 / 74.8 ±0.3
14		PET	**64.8** ±0.1	**86.9** ±0.2	**72.7** ±0.0	**85.3** ±0.2 / **85.5** ±0.4

Table 1: Average accuracy and standard deviation for RoBERTa (large) on Yelp, AG's News, Yahoo and MNLI (m:matched/mm:mismatched) for five training set sizes $|\mathcal{T}|$.

AutoPrompt

❑ 方法介绍

- 针对手动设计的离散提示模板的不稳定的问题
- AutoPrompt包含两步：自动搜索提示和自动搜索标签



Autoprompt: Eliciting knowledge from language models with automatically generated prompts. 2021.

AutoPrompt

❑ 实验结果

AutoPrompt在不同下游任务（如情感分析、事实检索）的表现优异

Model	Dev	Test
BiLSTM	-	82.8†
BiLSTM + ELMo	-	89.3†
BERT (linear probing)	85.2	83.4
BERT (finetuned)	-	93.5†
RoBERTa (linear probing)	87.9	88.8
RoBERTa (finetuned)	-	96.7†
BERT (manual)	63.2	63.2
BERT (AUTOPROMPT)	80.9	82.3
RoBERTa (manual)	85.3	85.2
RoBERTa (AUTOPROMPT)	91.2	91.4

Table 1: **Sentiment Analysis** performance

Prompt Type	Original			T-REx		
	MRR	P@10	P@1	MRR	P@10	P@1
LAMA	40.27	59.49	31.10	35.79	54.29	26.38
LPAQA (Top1)	43.57	62.03	34.10	39.86	57.27	31.16
AUTOPROMPT 5 Tokens	53.06	72.17	42.94	54.42	70.80	45.40
AUTOPROMPT 7 Tokens	53.89	73.93	43.34	54.89	72.02	45.57

Table 4: **Factual Retrieval:** On the left, we evaluate BERT on fact retrieval

AutoPrompt

❑ 实验结果

AutoPrompt学到的提示在人类看来不具备可读性，却能显著提升模型性能

Task	Prompt Template	Prompt found by AUTOPROMPT	Label Tokens
Sentiment Analysis	{sentence} [T]. . . [T] [P].	unflinchingly bleak and desperate Writing academicswhere overseas will appear [MASK].	**pos**: partnership, extraordinary, ##bla **neg**: worse, persisted, unconstitutional
NLI	{prem}[P][T]. . . [T]{hyp}	Two dogs are wrestling and hugging [MASK] concretepathic workplace There is no dog wrestling and hugging	**con**: Nobody, nobody, nor **ent**: ##found, ##ways, Agency **neu**: ##ponents, ##lary, ##uated
Fact Retrieval	*X plays Y music* {sub}[T]. . . [T][P].	Hall Overton fireplacemade antique son alto [MASK].	
Relation Extraction	*X is a Y by profession* {sent}{sub}[T]. . . [T][P].	Leonard Wood (born February 4, 1942) is a former Canadian politician. Leonard Wood gymnasium brotherdicative himself another [MASK].	

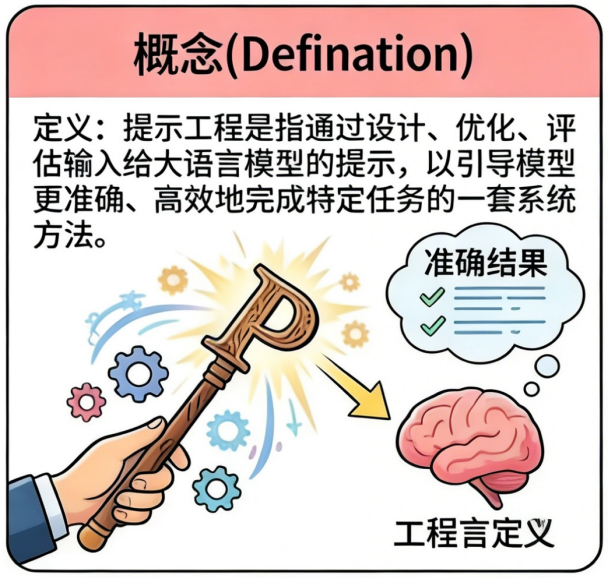
Table 3: **Example Prompts** by AUTOPROMPT for each task. On the left, we show the prompt template, which combines the input, a number of trigger tokens [T], and a prediction token [P]. For classification tasks (sentiment

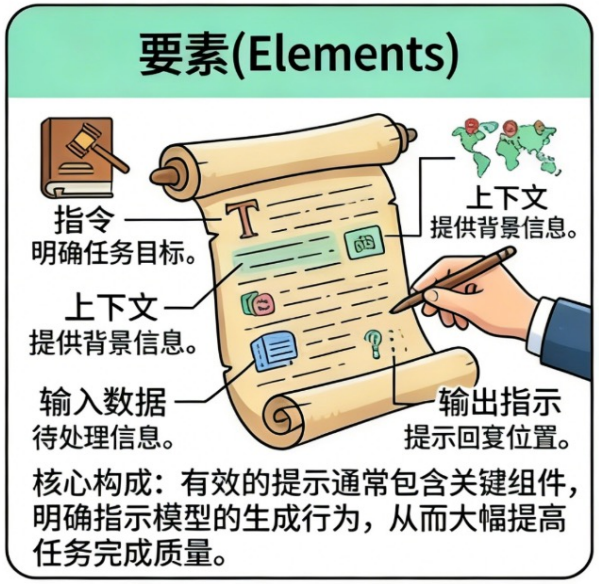
中国科学院大学
University of Chinese Academy of Sciences

目 录

提示工程

- 随着大模型的出现，提示工程的重要性显著提升，通过精心构造指令、示例与推理链，有效激发模型的理解与推理能力





提示工程

- 提示工程的组成：指令、上下文、输入数据、输出指示
  - 指令（Instruction）：明确告诉大模型需要完成的任务，比如“总结这段文本”
  - 上下文（Context）：可选，提供模型理解任务背景的必要信息
  - 输入数据（Input Data）：可选，提供模型关键的数据信息
  - 输出指示（Output Indicator）：可选，提示模型生成回复的位置

例子：请根据用户对电影的评论判断其情绪，并分类为积极、中性或消极。电影评论可能包含对剧情、表演或特效的评价，需要抓住整体态度来判断情绪倾向。“这部电影特效很震撼，剧情紧凑，但是角色塑造略显单薄，看完后感觉还行。”情感分析结果：

提示词编写指南

- 提示质量是高效使用大语言模型的关键
- 为提升提示的效果，可遵循以下基础原则：
  - 使用清晰、明确的语言描述问题
  - 在必要时提供充分的背景信息
  - 避免模糊、歧义或含糊不清的表达
- **推荐提示教程：**
  - Awesome ChatGPT Prompts https://github.com/f/prompts.chat
  - Prompting Guide中文版 https://www.promptingguide.ai/zh

提示工程的分类

- 按照示例数量划分
  - **Zero-shot**：无示例，仅任务描述
  - **One-shot**：单个示例
  - **Few-shot**：少量示例（典型方式）

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ← examples
4  plush girafe => girafe peluche      ← examples
5  cheese =>                           ← prompt
```

上下文学习

Zero-shot

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

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

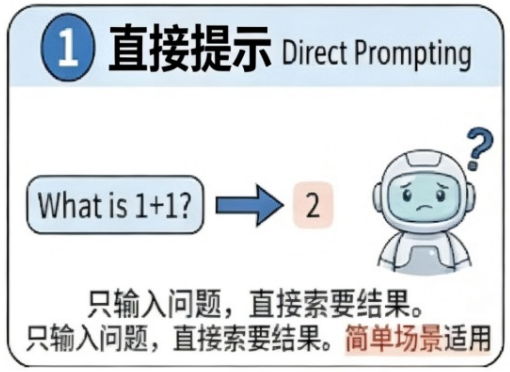
One-shot

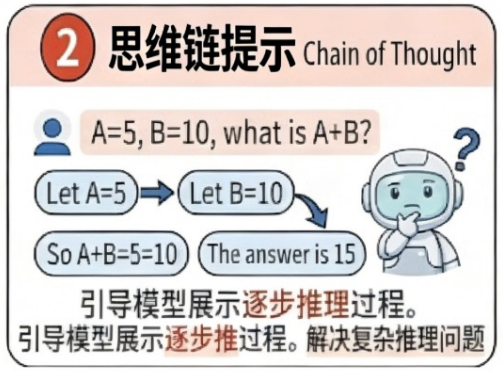
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

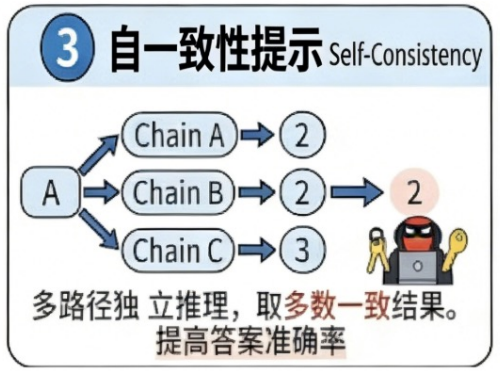
```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

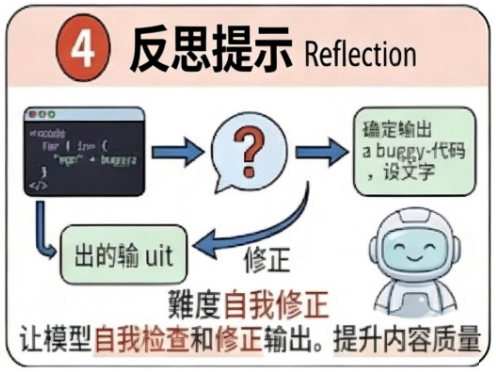
提示工程的分类

- 按照推理方式划分
  - **直接提示（Direct Prompting）**：直接给问题和结果
  - **思维链提示（Chain-of-Thought, CoT）**：引导模型逐步推理
  - **自一致性（Self-Consistency）**：多路径推理取多数结果
  - **反思提示（Reflection）**：让模型自我检查和修正









中国科学院大学
University of Chinese Academy of Sciences

目 录

上下文学习的兴起

- GPT-3 最早提出上下文学习（In-Context Learning，ICL），在Zero-shot、Few-shot场景下证明了有效性

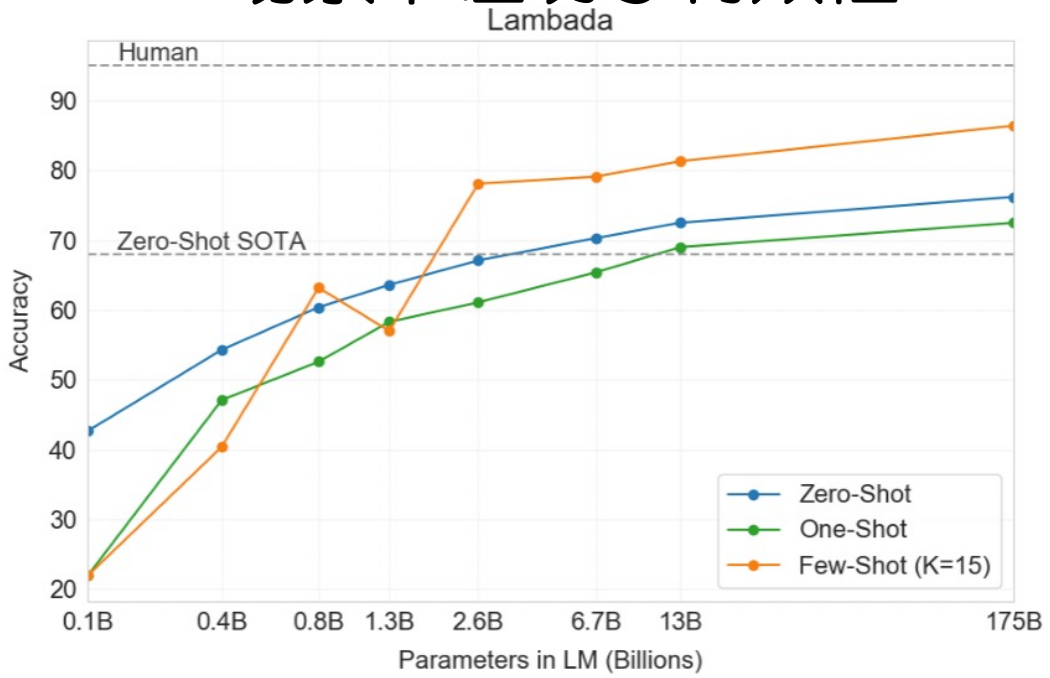
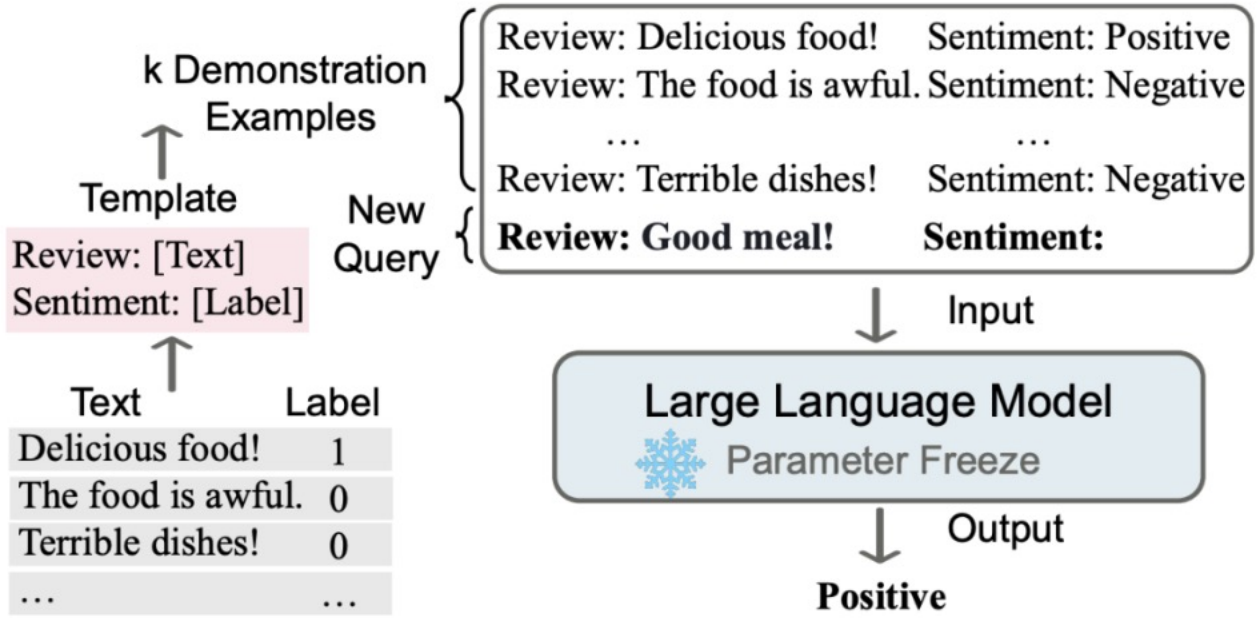


Figure 3.2: On LAMBADA, the few-shot capability of language models results in a strong boost to accuracy. GPT-3 2.7B outperforms the SOTA 17B parameter Turing-NLG [Tur20] in this setting, and GPT-3 175B advances the state of the art by 18%. Note zero-shot uses a different format from one-shot and few-shot as described in the text.

Language Models are Few-Shot Learners. 2020.

什么是ICL?

- 在推理阶段，大语言模型仅通过输入提示中的**少量演示示例**，**无需更新模型参数**，即可完成下游任务



A Survey on In-context Learning. 2023.

ICL的优势与局限

优势 (Advantages)

- **无需参数更新：**零计算成本，即插即用
- **灵活性高：**可快速切换任务，适应新场景
- **强大的泛化能力：**零样本/少样本能力强，是大模型智能的体现

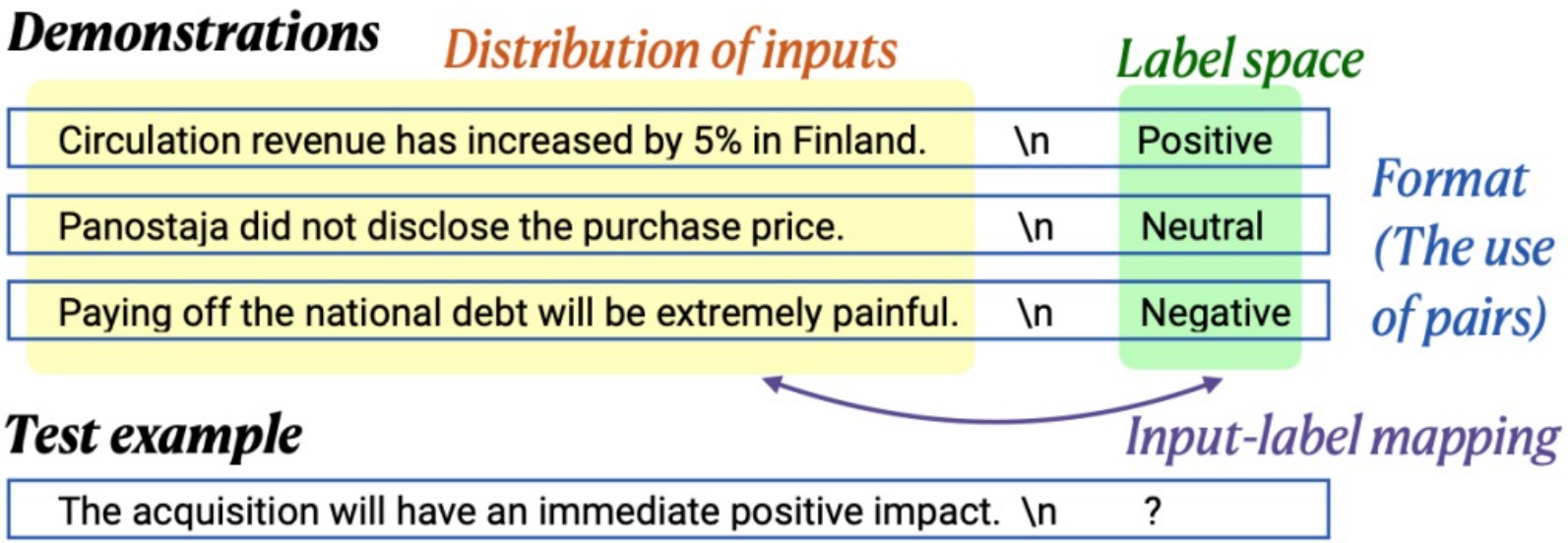
局限 (Limitations)

- **对示例敏感：**效果不稳定，受示例质量和数量影响大
- **推理过程不可见：**如何从示例中学习是一个“黑箱”
- **复杂任务表现有限：**在需要深度推理的任务上能力不足

总结：上下文学习是大模型的基础能力，虽然灵活高效，但也存在不可靠和黑箱问题，这正是后续技术（如思维链）试图解决的方向。

ICL核心要素

- 核心要素包括示例数量、示例相关性、输入文本分布、标签空间、输入-标签映射、使用格式等

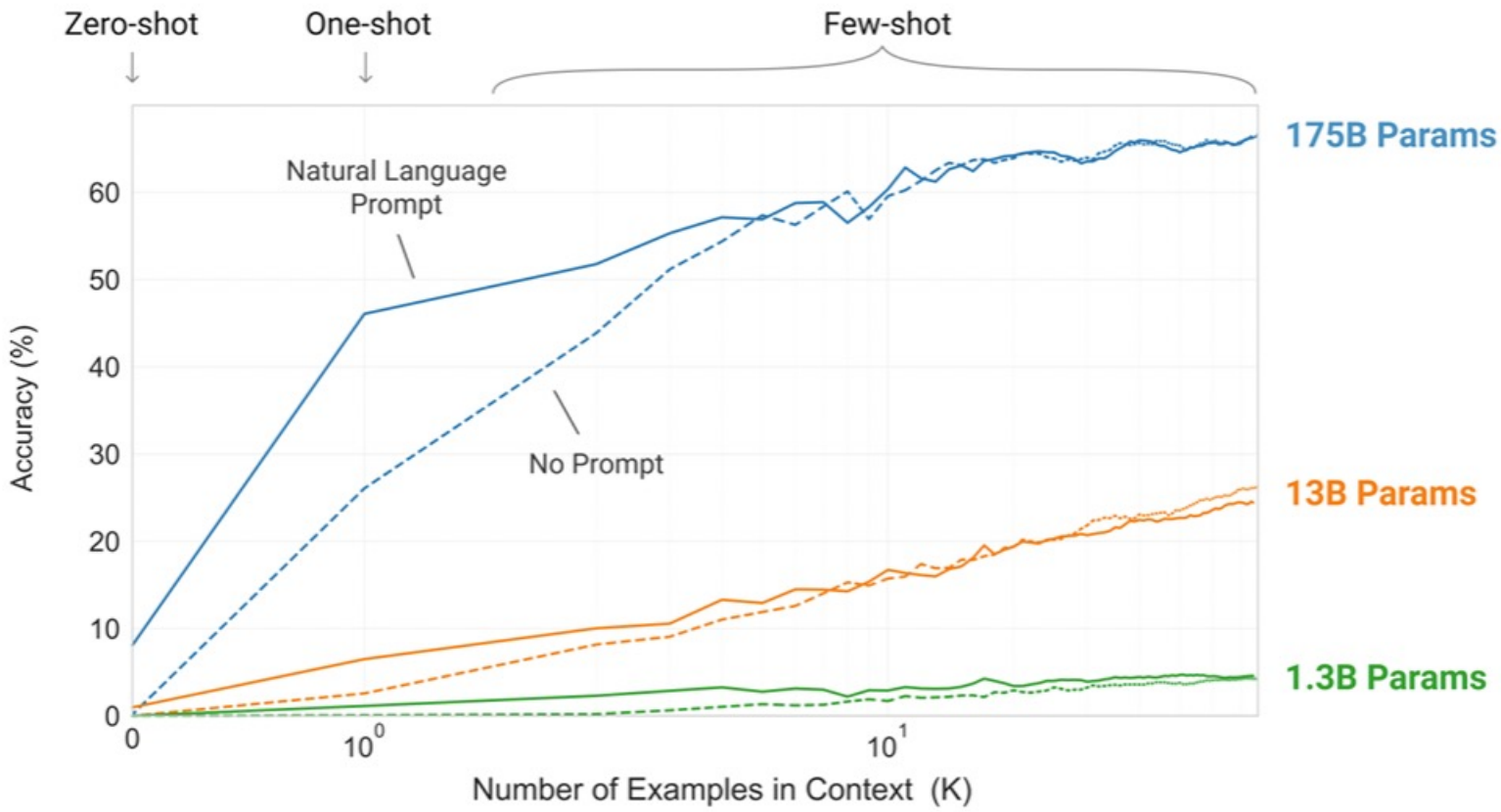


An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.

ICL核心要素

❑ **1. 示例数量**：提示中提供的样例数量

通常从 zero-shot → few-shot → many-shot，性能逐步提升，但过多也可能引入噪声



Language Models are Few-Shot Learners. 2020.

ICL核心要素

❑ **2. 示例相关性**：示例与当前输入的语义相似性

示例：

- **低相关示例**（电影评论 → 餐厅评论）

The plot is boring → Negative

- **高相关示例**

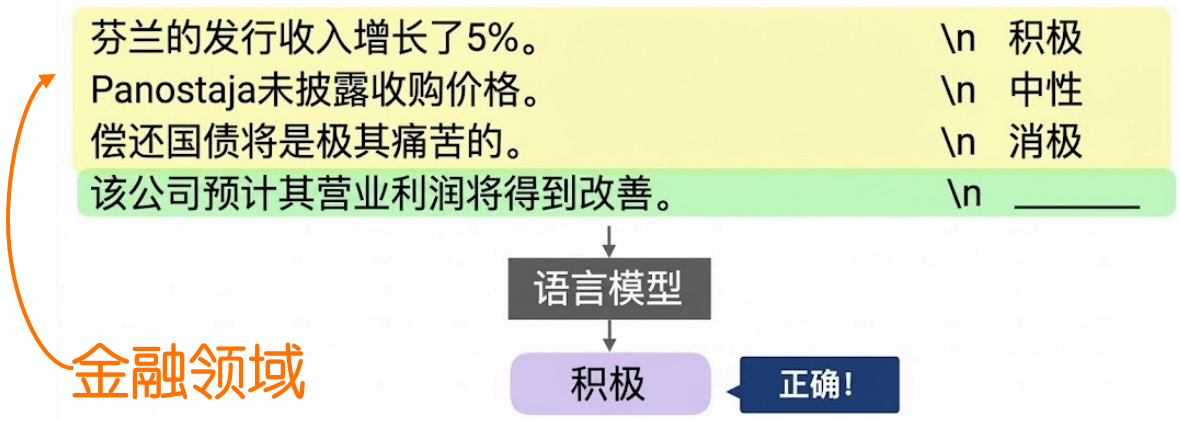
The food is terrible → Negative

👉 对“餐厅评论”，后者更有帮助。

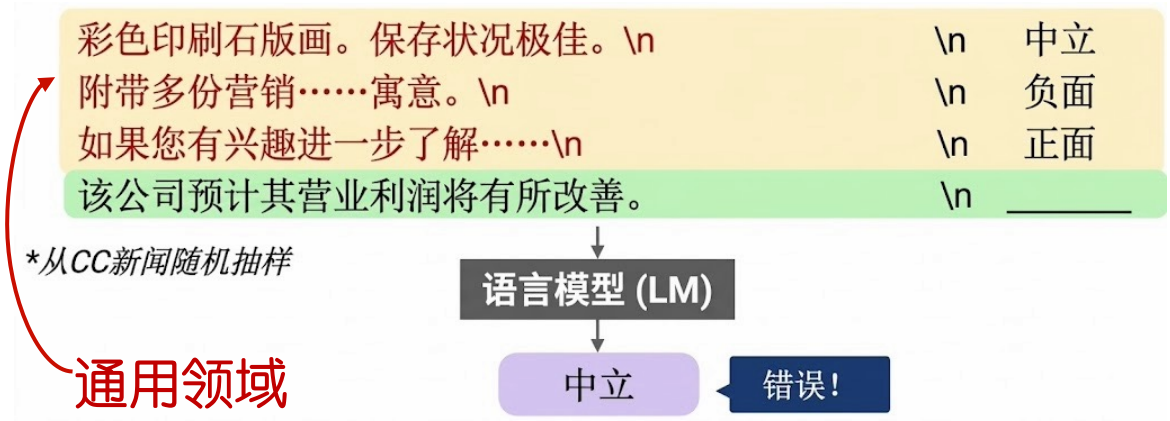
ICL核心要素

❑ **3. 输入文本分布**：包括领域分布、语言风格、文本复杂度等，示例数据的分布应尽量接近测试输入

(1) 相同领域分布的提示



(2) 不同领域分布的提示



Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. 2022.

ICL核心要素

❑ **4. 标签空间**：标签的设计（数量、粒度、表达方式），标签空间应该具有更易理解的含义

示例 1（粒度）：

- **简单标签**

Positive / Negative

- **细粒度标签**

Positive / Neutral / Negative

👉 标签越细，任务越难但更精准

示例2（表达方式）：

- **数值标签**

1 / 0

- **自然语言标签**

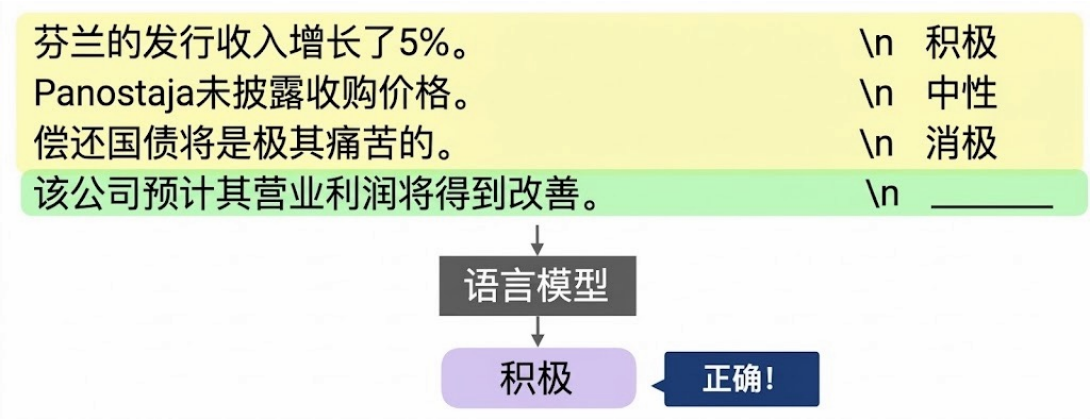
是/ 不是

👉 标签应该尽量采用自然语言形式

ICL核心要素

❑ **5. 输入-标签映射：** 输入与标签的映射关系是否清晰、一致

(1) 带有真实标签输出的提示



(2) 带有随机标签输出的提示

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. 2022.

ICL核心要素

❑ **6. 使用格式**：示例的组织形式（如自然语言、表格、JSON等）

示例：

- **自然语言格式**

  输入文本：I love it

  情感标签：Positive

- **JSON格式**

  {“文本文本”: “I love it”, “情感标签": "Positive"}

👉 某些任务（如信息抽取）用结构化格式更稳定。

ICL如何运作?

- 模型如何在没有训练的情况下“学习”解决新任务（例如，通过梯度更新来优化模型在新任务上的性能）？

ICL机理分析：

隐式贝叶斯推理假设

An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.
Transformers Learn In-Context by Gradient Descent. 2022.

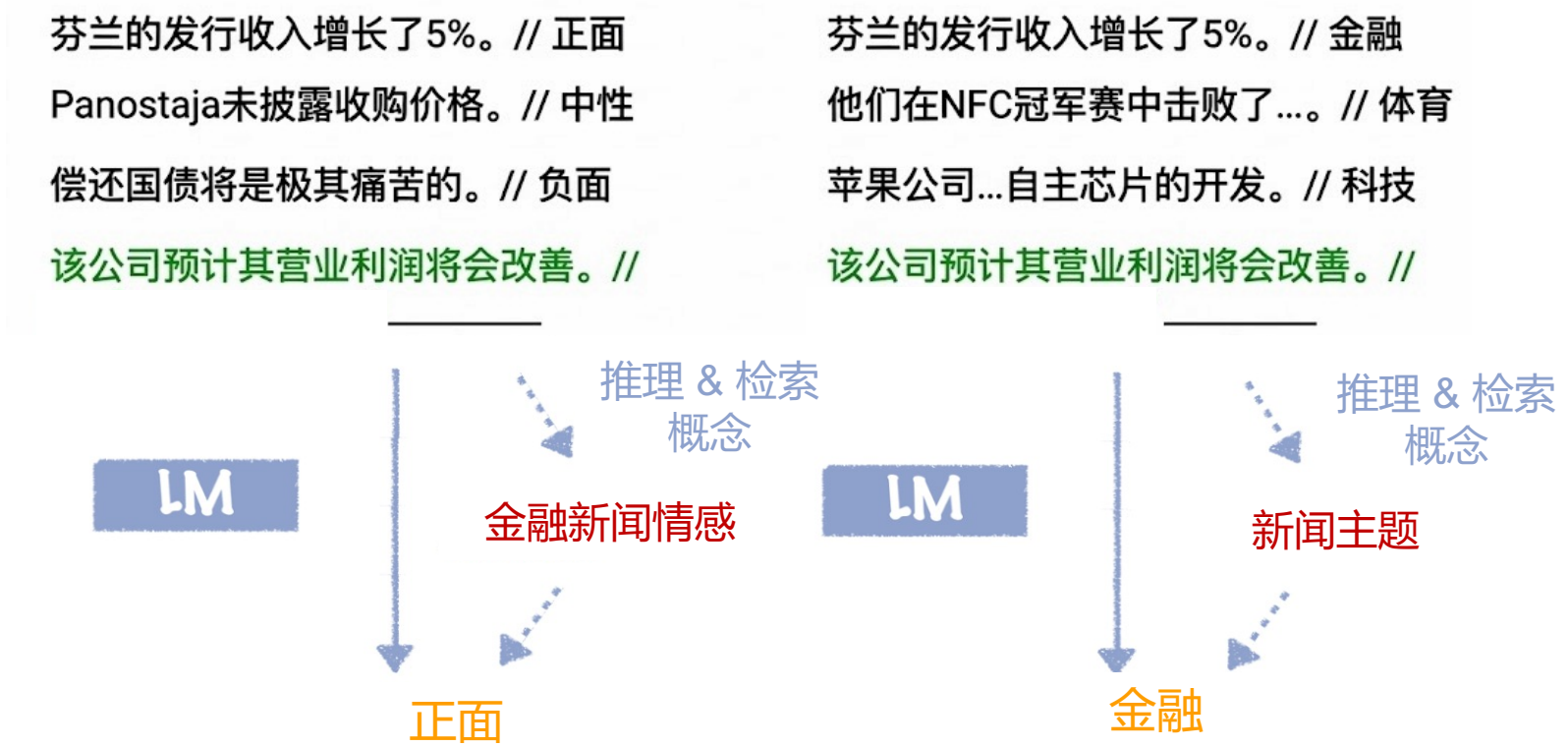
ICL-隐式贝叶斯推理假设

- 给定三个示例（同输入，不同标签），模型如何进行判断?

An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.

ICL-隐式贝叶斯推理假设

- 根据提供的示例，模型会**判断任务是情感分析(左)还是主题分类(右)**，并将同一套映射规则应用到测试输入

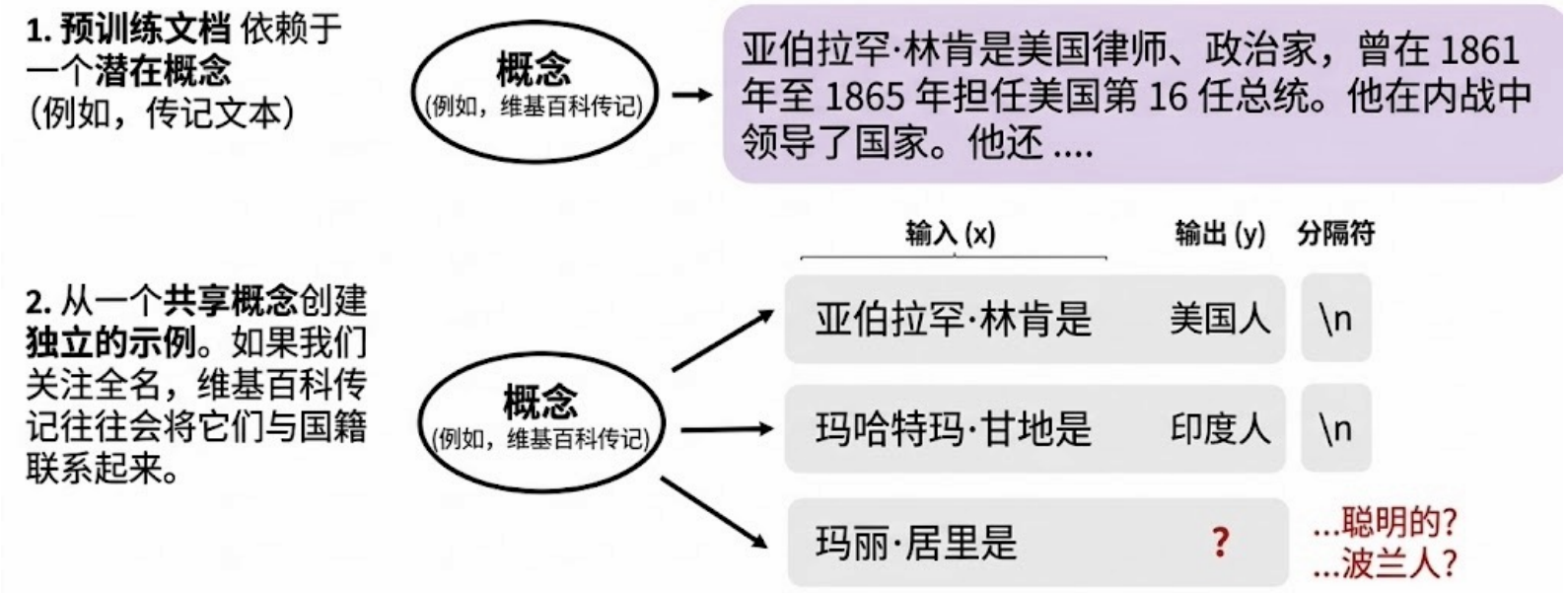


这个过程称为**学习潜在概念(concept)**

An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.

ICL-隐式贝叶斯推理假设

- 在预训练阶段，模型通过文本语料学会了建模海量潜在概念
- 在推理阶段，模型根据提示词中示例，**来定位预训练中习得的某些潜在概念**，以此完成新任务



An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.

ICL-隐式贝叶斯推理假设

- 概念（concept）：包含各种统计特征的**隐变量**
- 比如“新闻主题”概念包含：
  - 词汇分布（新闻及其主题）
  - 文本格式（新闻文章的写作方式）
  - 新闻与主题的关联
  - 以及其他语义与句法关系等

An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.

ICL-隐式贝叶斯推理假设

- “定位潜在概念”过程 可看作**基于提示词共享概念的贝叶斯推理**。
  模型若能推断出该概念，就能对测试示例做出正确预测

$$p(\text{output}|\text{prompt}) = \int_{\text{concept}} p(\text{output}|\text{concept}, \text{prompt}) p(\text{concept}|\text{prompt}) d(\text{concept}).$$

金融

新闻主题

芬兰的发行收入增长了5%。// 金融

他们在NFC冠军赛中击败了...。// 体育

苹果公司...自主芯片的开发。// 科技

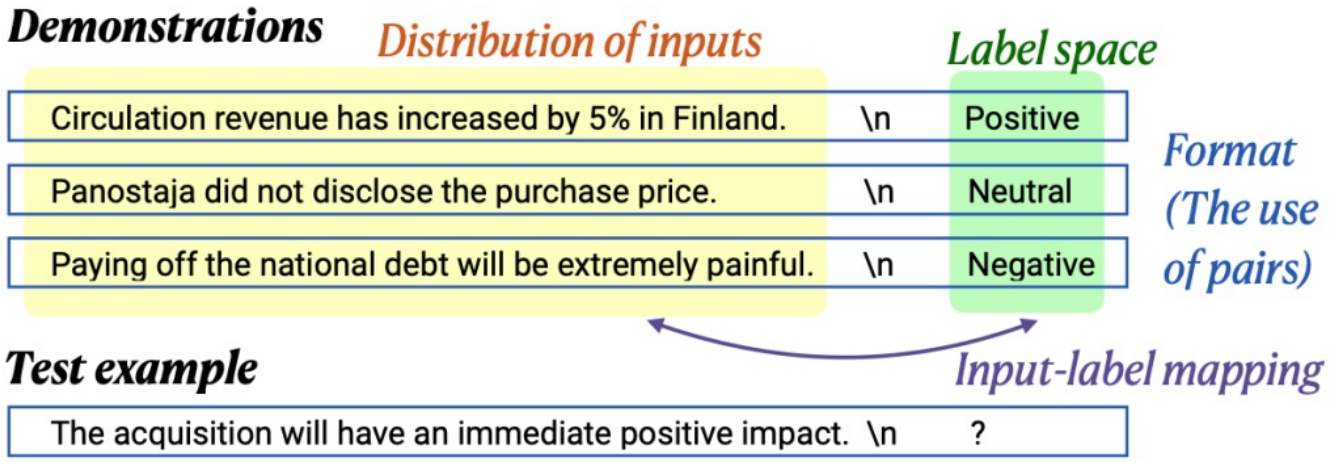
该公司预计其营业利润将会改善。//

An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.

隐式贝叶斯推理假设：实验效果

❑ 从四个维度探究ICL的效果增益

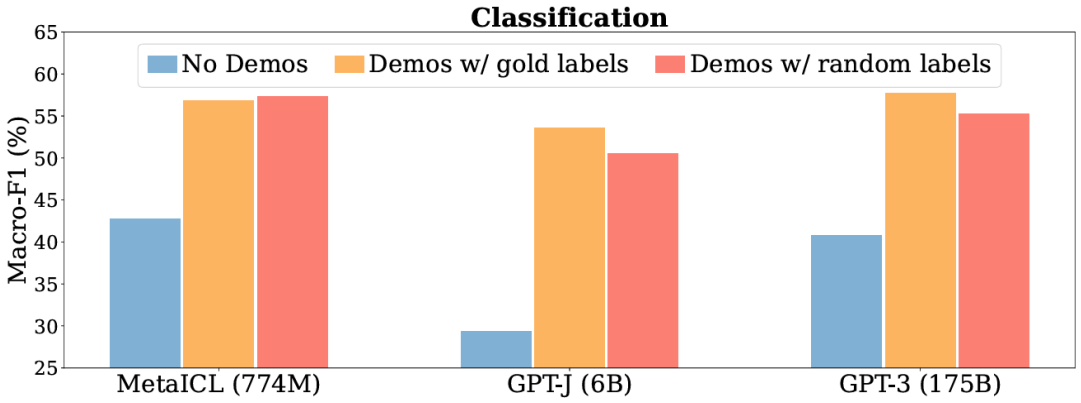
- 输入-标签映射
- 输入分布
- 标签空间
- 提示格式

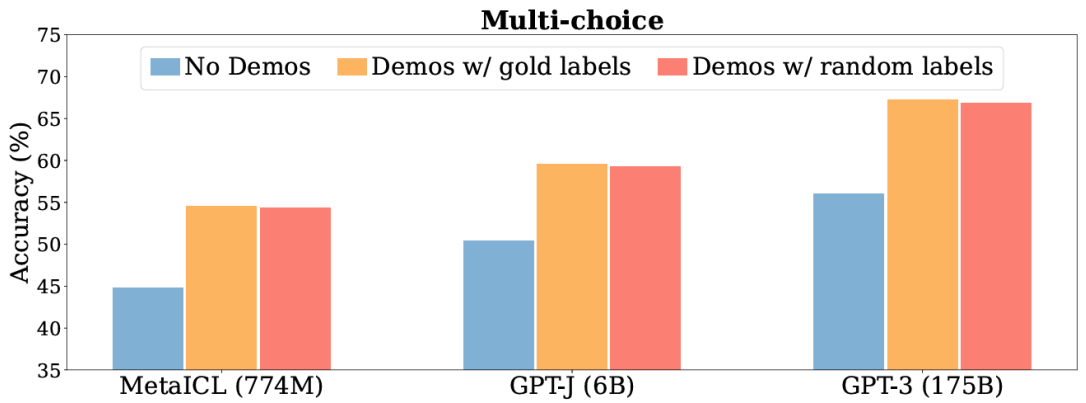


Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. 2022.

隐式贝叶斯推理假设：实验效果

- **结论1：**ICL中提示词中输入 - 输出对的重要性远低于预期
  - No Demos：仅输入测试文本
  - Demos w/ gold labels：提示词含真实输入 - 输出对（标准上下文学习）
  - Demos w/ random labels：提示词输入不变，输出从输出集合随机采样





👉 真实标签替换为随机标签，只会对性能造成轻微影响。 反直觉！

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. 2022.

隐式贝叶斯推理假设：实验效果

❑ **结论2：**ICL中使用分布外数据（OOD）会导致模型性能下降

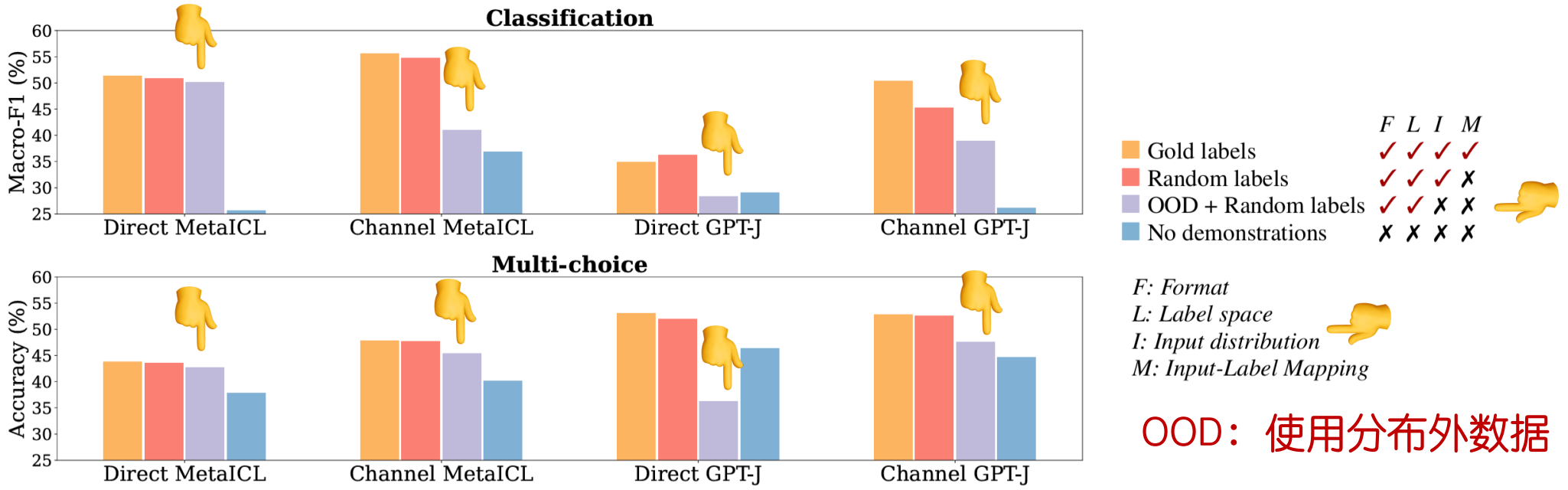
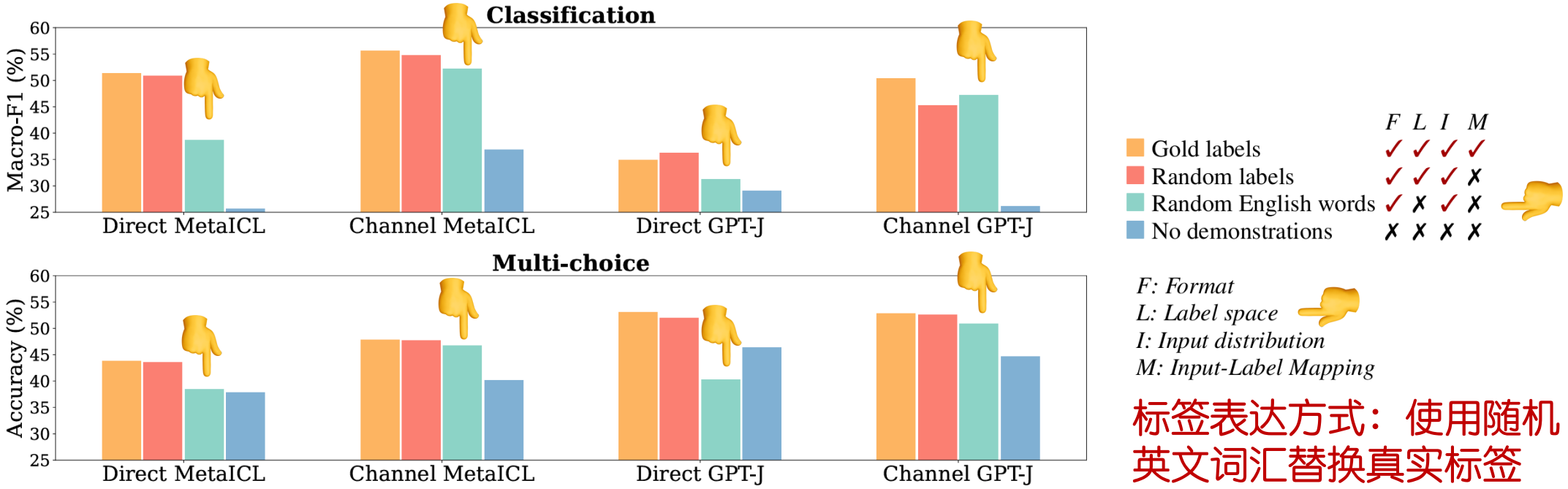


Figure 8: Impact of the distribution of the inputs. Evaluated in classification (top) and multi-choice (bottom). The impact of the distribution of the input text can be measured by comparing ■ and ■. The gap is substantial, with an exception in Direct MetaICL (discussion in Section 5.1).

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. 2022.

隐式贝叶斯推理假设：实验效果

❑ **结论3**：标签空间的一致性显著有助于提高性能



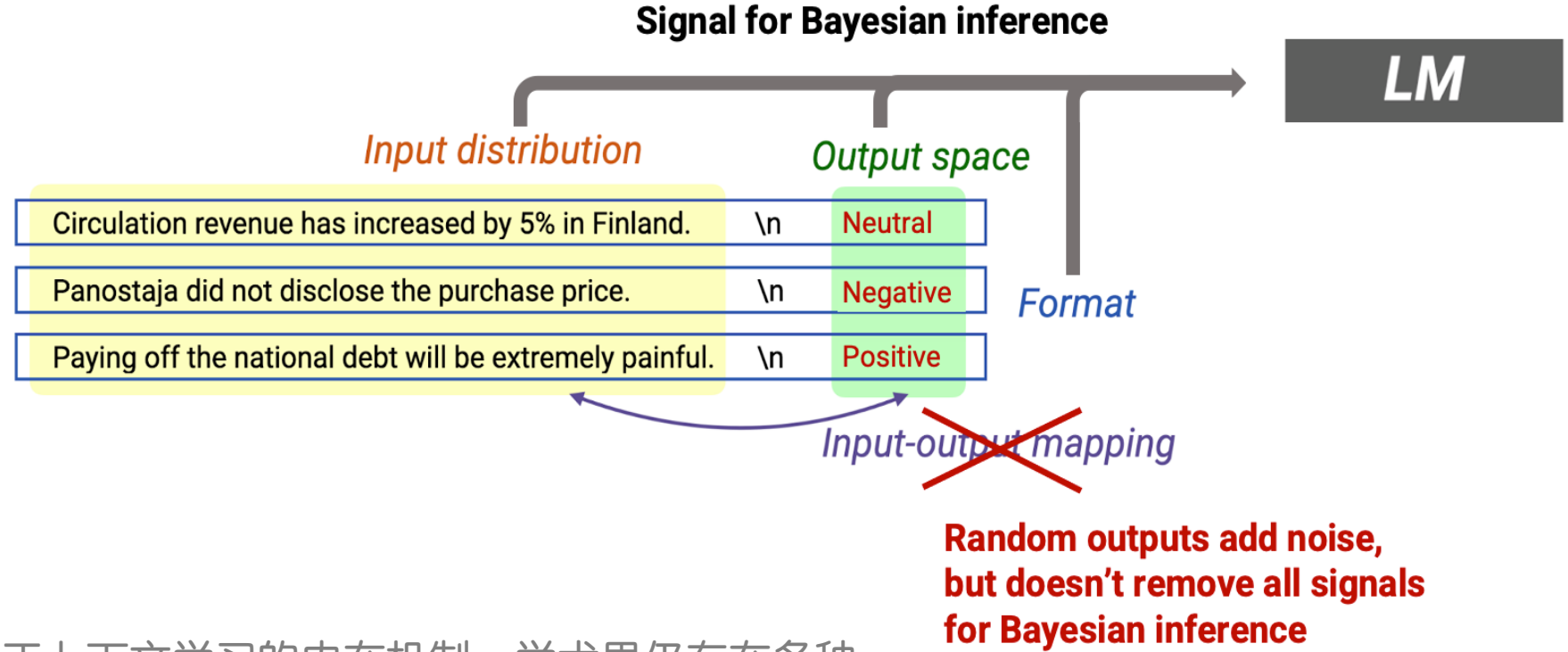
标签表达方式：使用随机英文词汇替换真实标签

Figure 9: Impact of the label space. Evaluated in classification (top) and multi-choice (bottom). The impact of the label space can be measured by comparing ■ and ■. The gap is significant in the direct models but not in the channel models (discussion in Section 5.2).

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. 2022.

隐式贝叶斯推理假设：实验结论

- ICL的性能提示主要来源于：输入分布、标签空间和提示格式



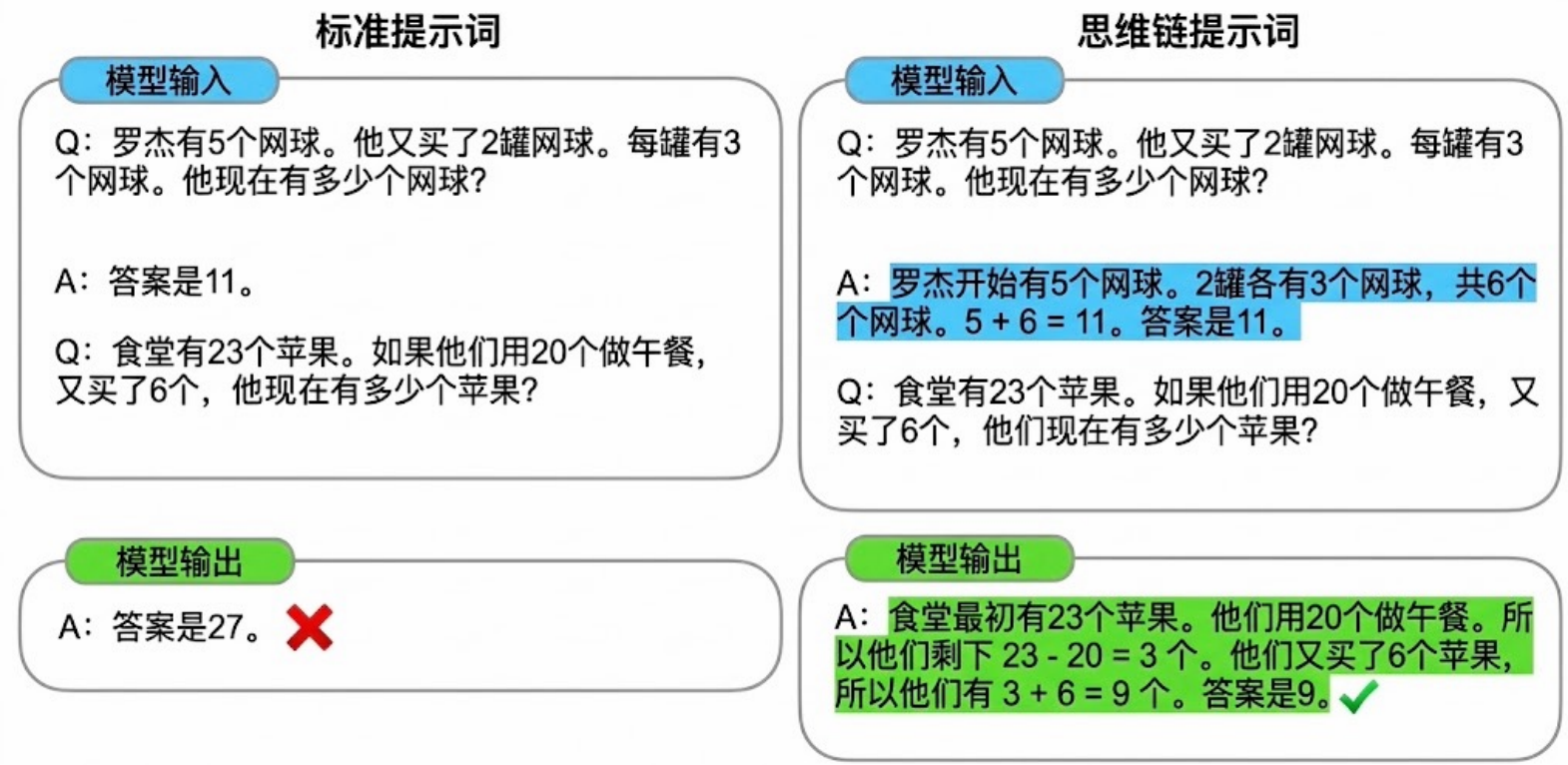
注：关于上下文学习的内在机制，学术界仍存在多种不同观点与解释路径，尚未形成统一结论。

https://ai.stanford.edu/blog/understanding-incontext/

目 录

思维链

❑ 思维链（Chain of Thought，CoT）通过**显式生成中间推理步骤**，以增强大语言模型在复杂任务中的推理能力



Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. 2022.

思维链示例

❑ 更多示例

数学应用题（自由回答）

Q: 罗杰有5个网球。他又买了2罐网球。每罐有3个网球。他现在有多少个网球?

A: 罗杰一开始有5个球。2罐，每罐3个，共6个球。5 + 6 = 11。答案是 11。

日期理解

Q: 音乐会原定于 1943年6月1日举行，但推迟了一天到今天。10天前的日期是几月几日 (MM/DD/YYYY)?

A: 1943年6月1日后的一天是 1943年6月2日，所以今天是 1943年6月2日。今天之前的10天是5月23日。所以答案是05/23/1943。

最后一个字母拼接

Q: 取 "Lady Gaga" 单词的最后一个字母并将其拼接。

A: "Lady" 的最后一个字母是 "y"。"Gaga" 的最后一个字母是 "a"。拼接它们是 "ya"。所以答案是 ya。

数学应用题（多项选择）

Q: 打出1到500的数字需要按多少次键盘?
选项: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: 1到9有9个一位数。10到99有90个两位数。100到500有401个三位数。9 + 90(2) + 401(3) = 1392。答案是（b）。

体育理解

Q: 以下句子是否合理? "若昂·穆蒂尼奥在NFC锦标赛中接住了屏幕传球。"

A: 若昂·穆蒂尼奥是一名足球运动员。NFC锦标赛是美式橄榄球的一部分，不是足球。所以答案是否。

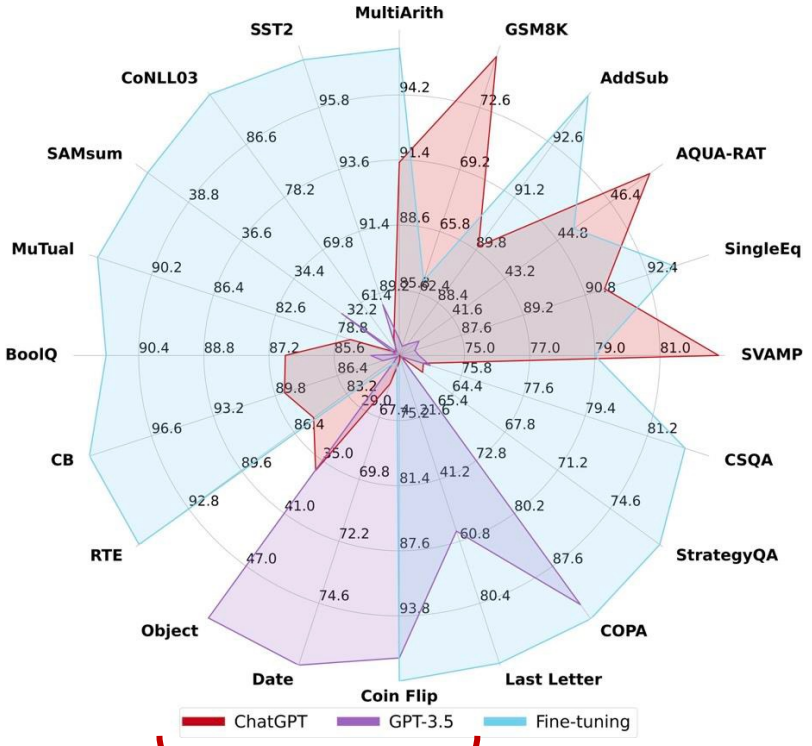
硬币翻转（状态跟踪）

Q: 硬币正面朝上。梅贝尔翻转了硬币。沙隆达没有翻转硬币。硬币还是正面朝上吗?

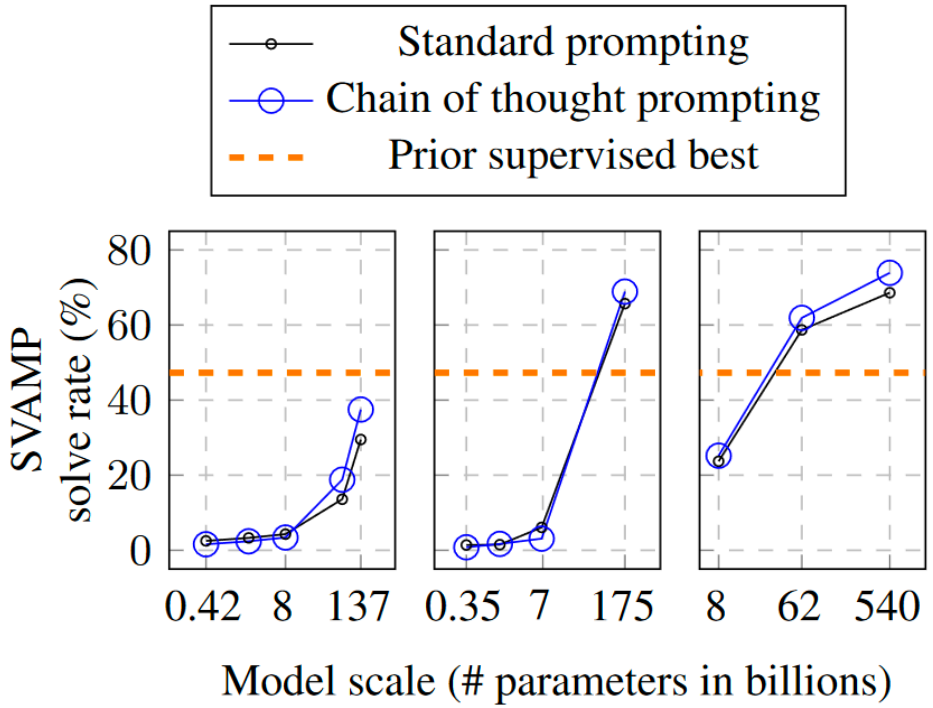
A: 硬币被梅贝尔翻转了。所以硬币被翻转了 1次，这是一个奇数。硬币开始时正面朝上，所以在奇数次翻转后，它将反面朝上。所以答案是否。

思维链效果

- 思维链在**数学题、逻辑推理、多步骤任务**等场景中表现突出



基于思维链提示



Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. 2022.

思维链的工作原理

❑ 思维链内在机理： 从“黑箱”到“白盒”

结构化剪枝器

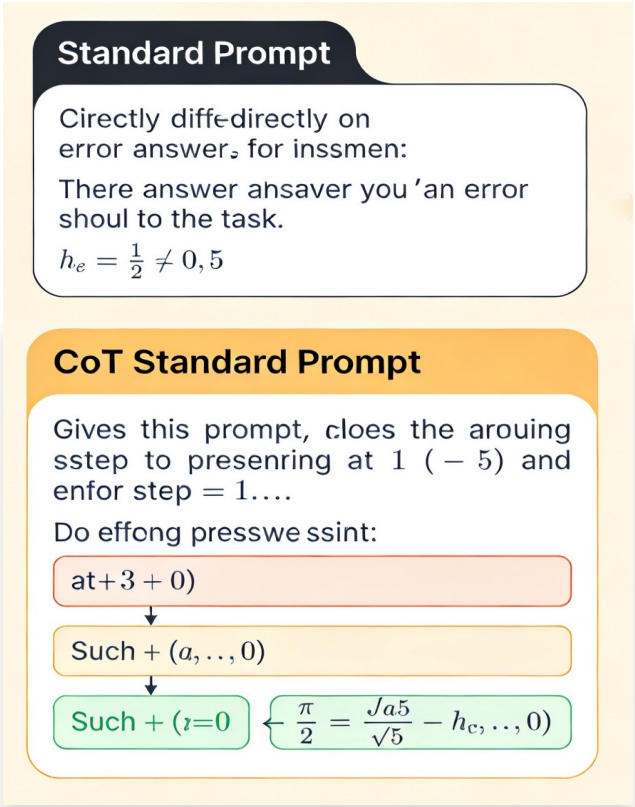
引导模型遵循特定的推理路径，约束输出空间，减少无关或错误的生成。

概率分布收敛

分步推理使模型对每一步的预测更加确定，概率分布更集中，从而提高准确性。

任务相关的神经元激活

激活模型中与任务推理相关的特定神经元，使其更专注于解决问题的关键特征。



思维链的核心优势

提升复杂推理准确性

通过将复杂问题拆解为可验证的中间步骤，降低错误累积

增强决策透明度与可解释性

打破传统AI“黑箱”特性，使推理过程显式化，增加模型的可解释性

便于错误定位与调试

通过检查中间步骤快速定位问题，而非面对无法追溯的最终结果

促进人机协作与信任

透明的推理过程建立了用户对AI决策的信任，使其愿意采纳AI建议

思维链的主要缺点与局限性

模型规模依赖性强

推理能力主要在超大模型（>100B参数）中才能有效“涌现”，中小模型难以显现

推理路径的可靠性存疑

可能生成看似合理但实际错误的推理链，甚至出现“步骤错误但答案巧合正确”的情况

计算成本与效率问题

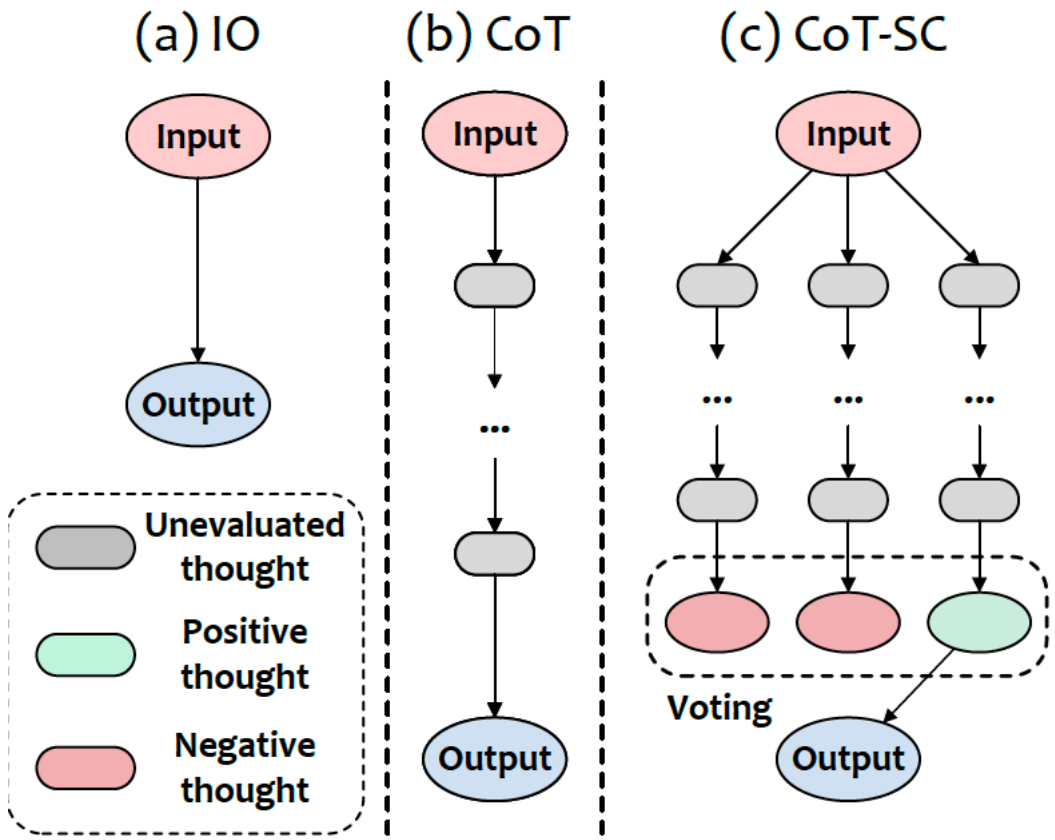
生成冗长的推理链会显著增加Token消耗和推理时间，导致计算成本上升和效率降低

对简单任务的适用性差

对于单步事实查询或创意生成等任务，CoT不仅毫无必要，还会引入冗余步骤，降低效率

常见CoT方法

- 零样本思维链
- 少样本思维链
- 自洽思维链
- 程序思维链
- 表格思维链
- …… ……



Zero-shot CoT

- **零样本思维链：**不提供示例，只加一句提示：**“Let’s think step by step.”（让我们一步步思考）**
- **少样本思维链：**在提示中加入带推理过程的示例

(b) Few-shot-CoT

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 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

(d) Zero-shot-CoT (Ours)

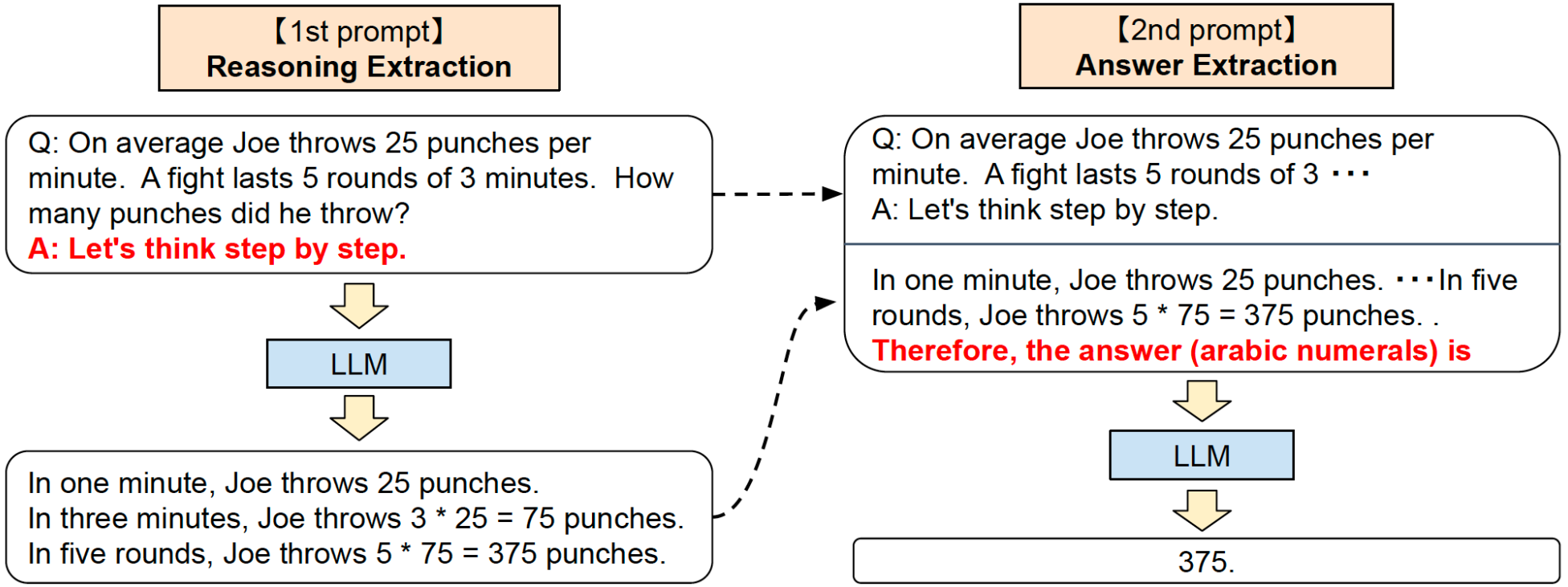
Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let’s think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Large Language Models are Zero-Shot Reasoners. 2022.

Zero-shot CoT

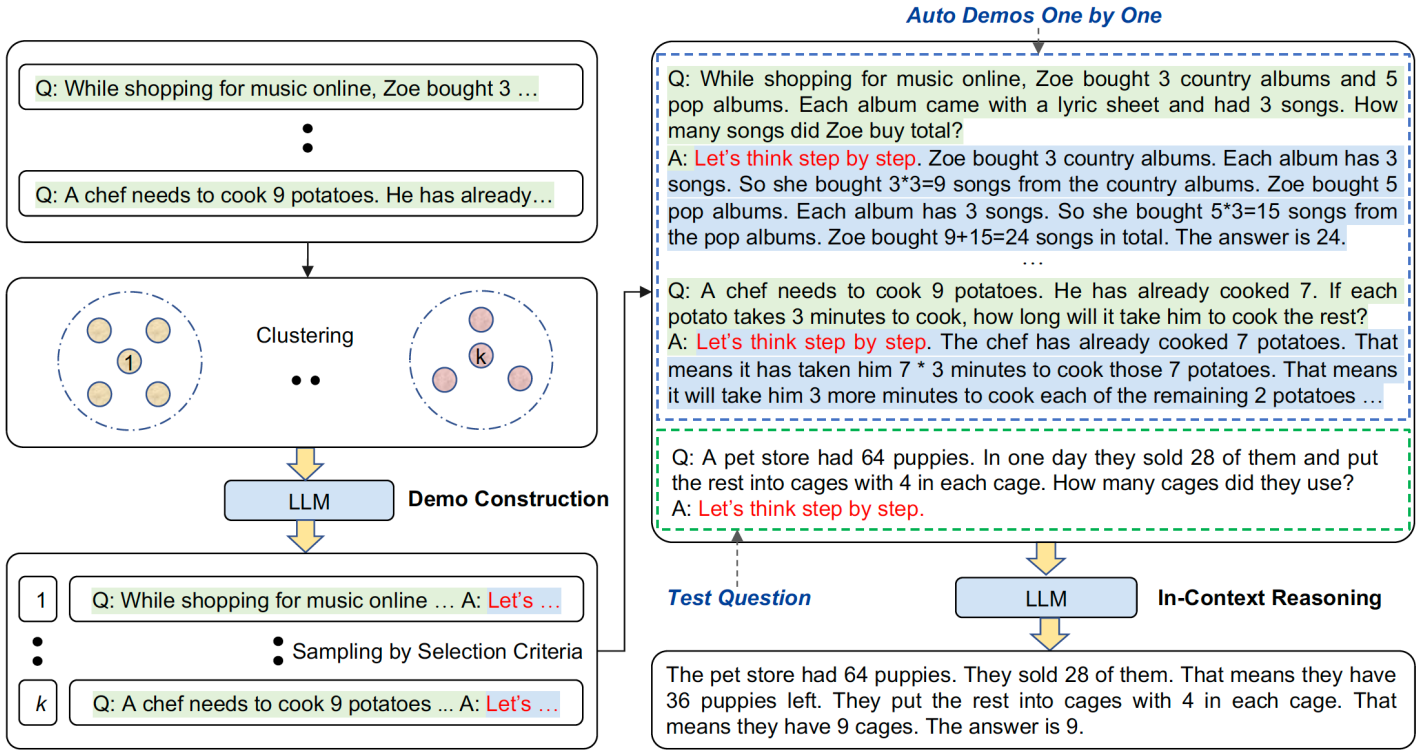
❑ **零样本思维链改进**：使用两个Prompt依次执行推理提取和答案提取



Large Language Models are Zero-Shot Reasoners. 2022.

Auto-CoT

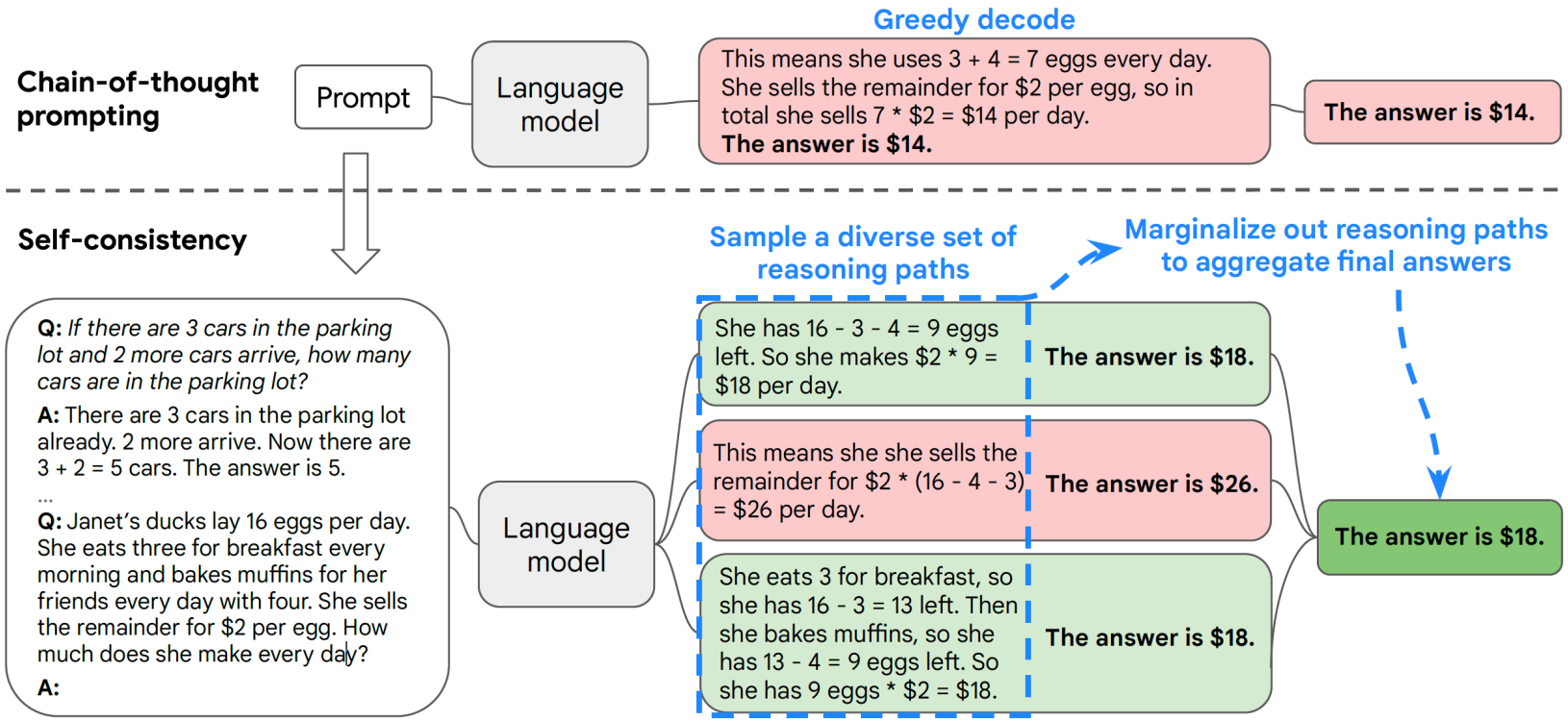
- **自动思维链（Auto-CoT）**：一种自动化的思维链提示生成方法
- 两步核心流程：
  1. **问题聚类**：将问题分成若干“簇”
  2. **示例抽样**：从每类选取代表问题，用 Zero-Shot CoT 生成推理链作为示例



Automatic Chain of Thought Prompting in Large Language Models. 2022.

CoT-SC

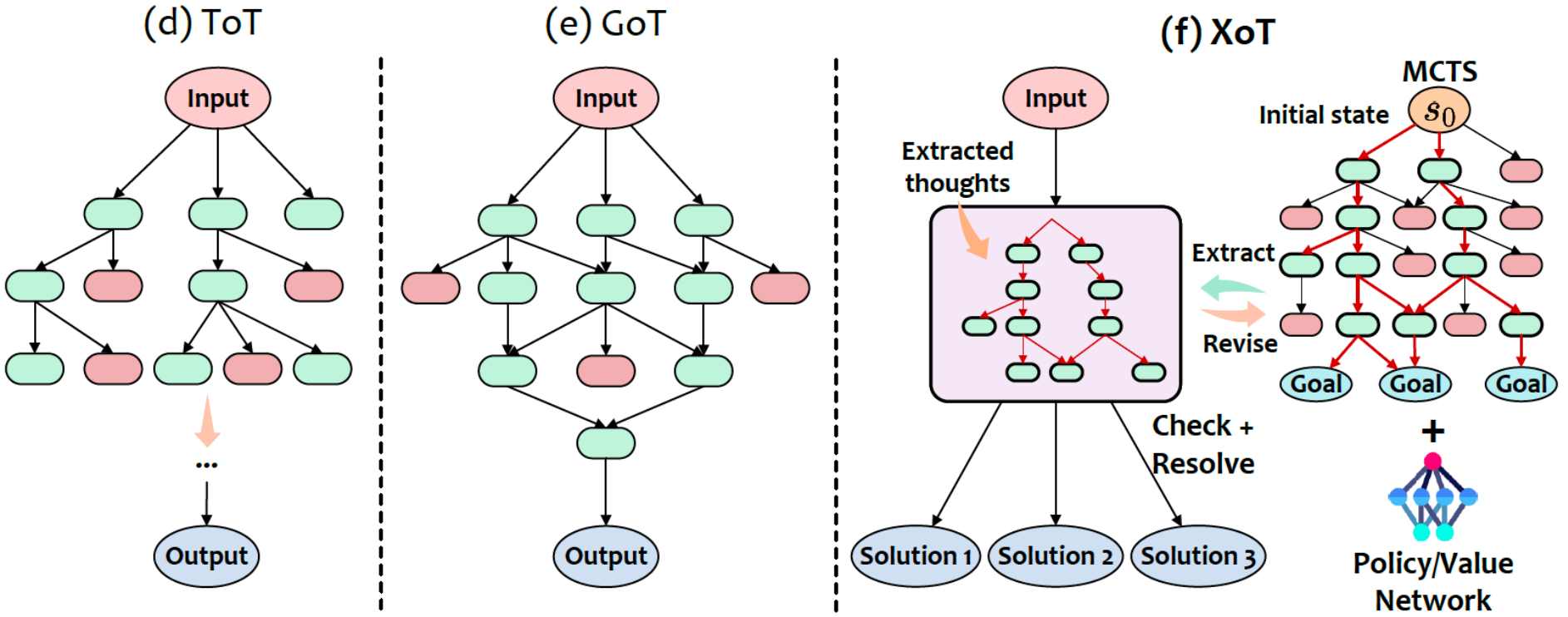
❑ **自洽思维链（Self Consistency with CoT）：** 基于思维链学习多次采样生成推理路径，并通过投票选择最一致的答案



Self-Consistency Improves Chain of Thought Reasoning in Language Models. 2022.

CoT结构变体：XoT系列

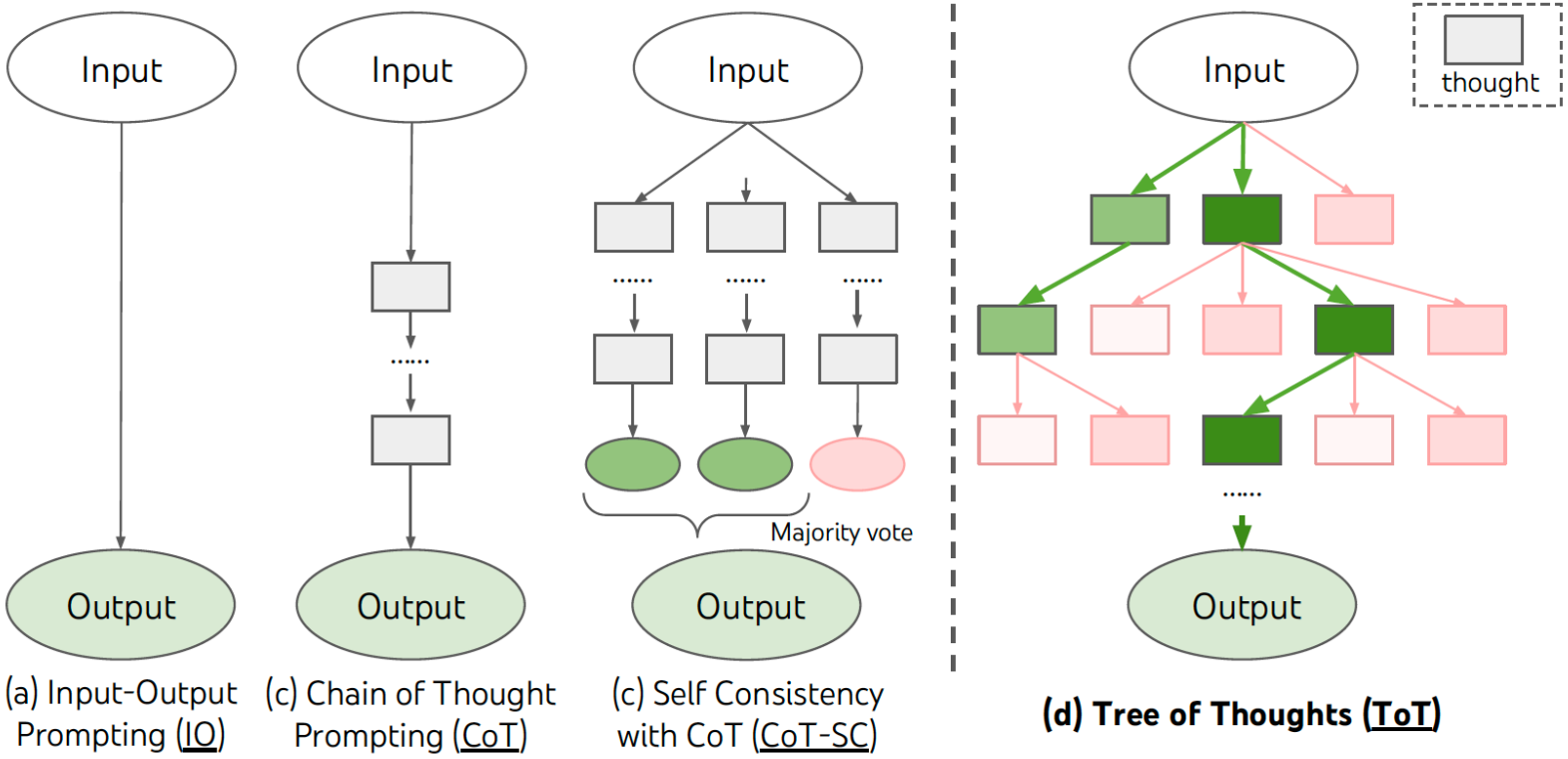
- 思维链的结构变体包括：思维树（Tree of Thought）、思维图（Graph of Thought）以及 XoT 等形式



Language Models are Few-Shot Learners

ToT

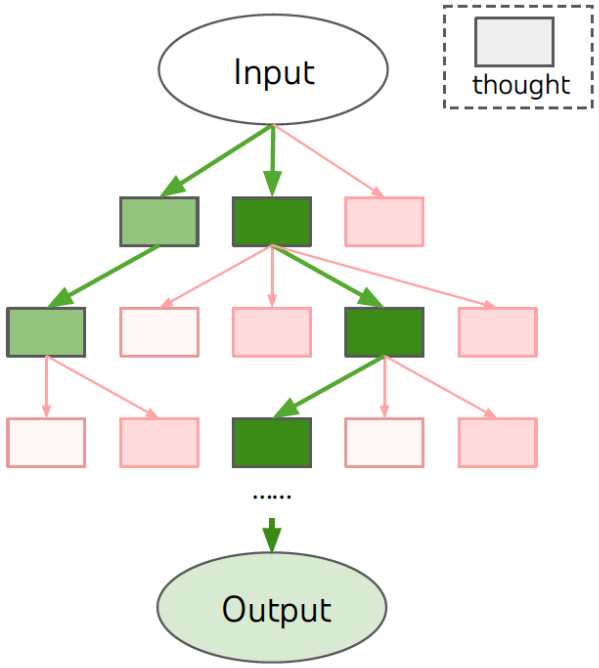
❑ 思维树（Tree-of-Thought, ToT）将思维过程组织为树状结构，并利用搜索算法（如BFS/DFS）进行扩展，以寻找最优解



Tree of Thoughts: Deliberate Problem Solving with Large Language Models. 2023.

ToT

❑ ToT 将任何问题建模为在树上的搜索，通常包括四个核心过程：



(d) Tree of Thoughts (ToT)

- **思维分解（Decomposition）**
  将中间推理过程拆解为多个思维步骤；
- **思维生成（Generation）：**
  在当前状态下产生多个候选思维；
- **状态评估（Evaluation）：**
  对每个状态进行打分或优劣判断；
- **搜索算法（Search）：**
  利用搜索算法，扩展最有潜力的思维树。

Tree of Thoughts: Deliberate Problem Solving with Large Language Models. 2023.

ToT

- 在 24 点游戏中，ToT 将问题分解为多个中间表达式，在每步生成多个候选思维并进行评估，最后通过 BFS 保留最优路径

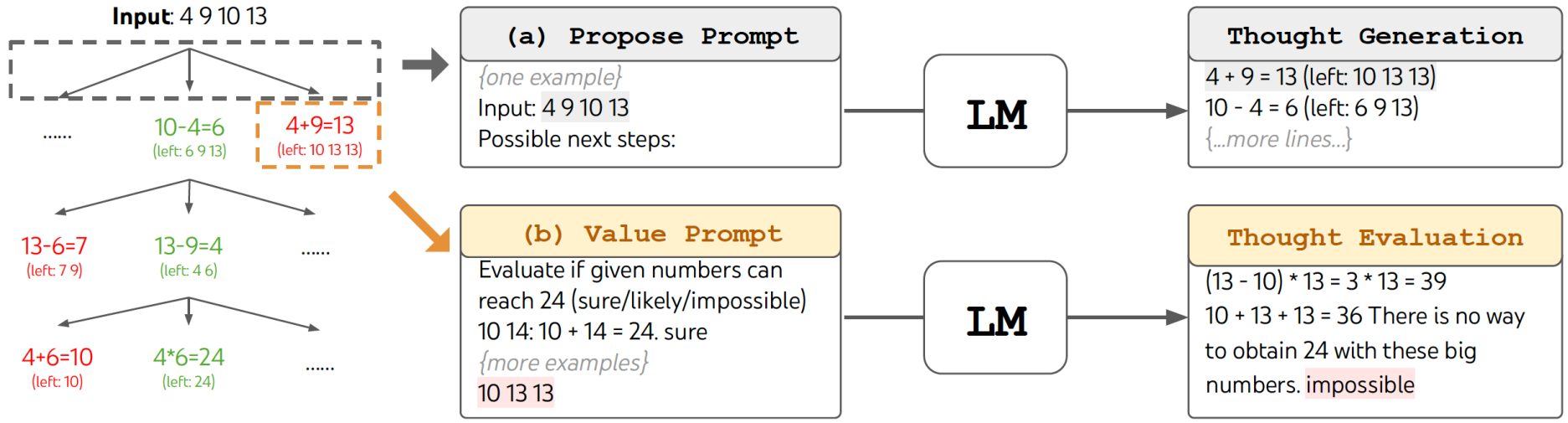


Figure 2: ToT in a game of 24. The LM is prompted for (a) thought generation and (b) valuation.

Tree of Thoughts: Deliberate Problem Solving with Large Language Models. 2023.

ToT

❑ 24点游戏实验结果

Method	Success
IO prompt	7.3%
CoT prompt	4.0%
CoT-SC (k=100)	9.0%
ToT (ours) (b=1)	45%
ToT (ours) (b=5)	**74%**
IO + Refine (k=10)	27%
IO (best of 100)	33%
CoT (best of 100)	49%

Table 2: Game of 24 Results.

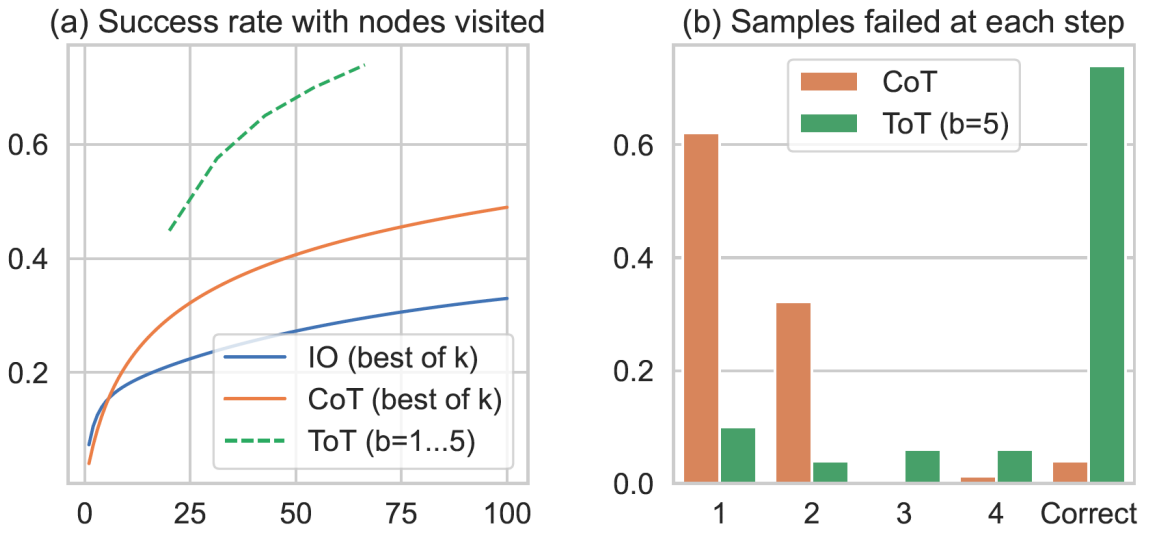


Figure 3: Game of 24 (a) scale analysis & (b) error analysis.

通过在思维树中探索多条路径，并进行状态评估，ToT 能避免早期错误，明显优于单路径解码的 CoT。

Tree of Thoughts: Deliberate Problem Solving with Large Language Models. 2023.

GoT

- 思维图（Graph-of-Thought, GoT）：在 ToT 的基础上进行了扩展，将推理过程建模为图结构，允许不同节点的交互

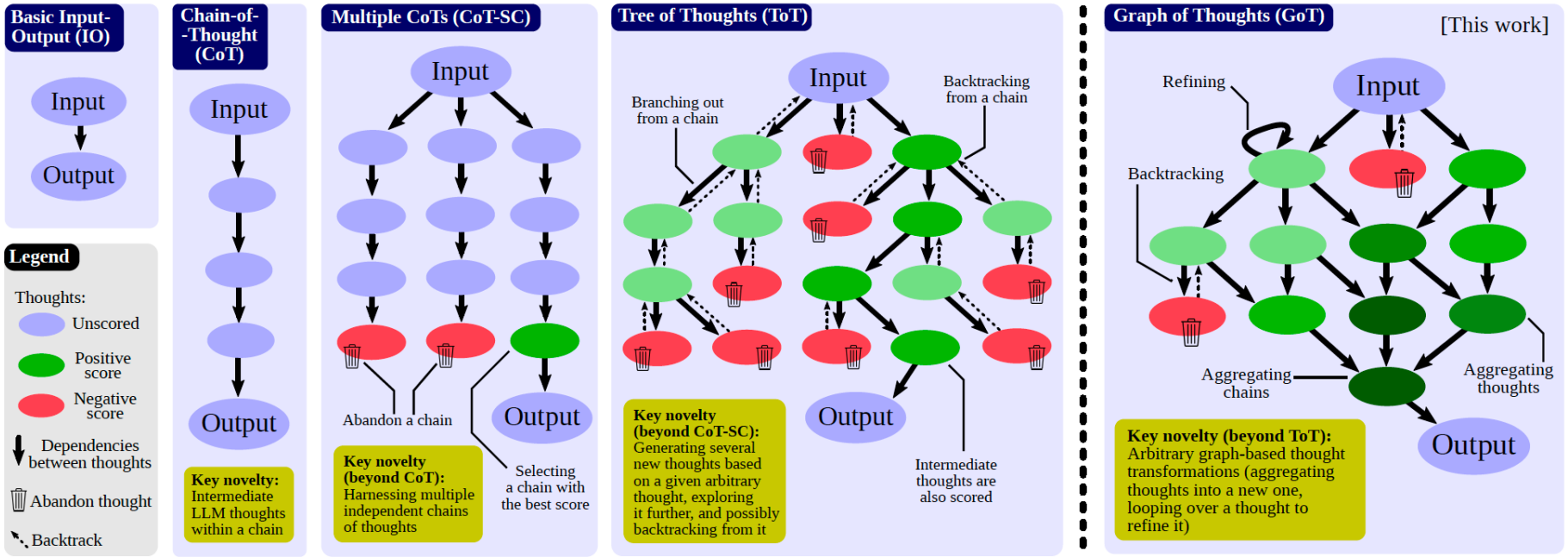
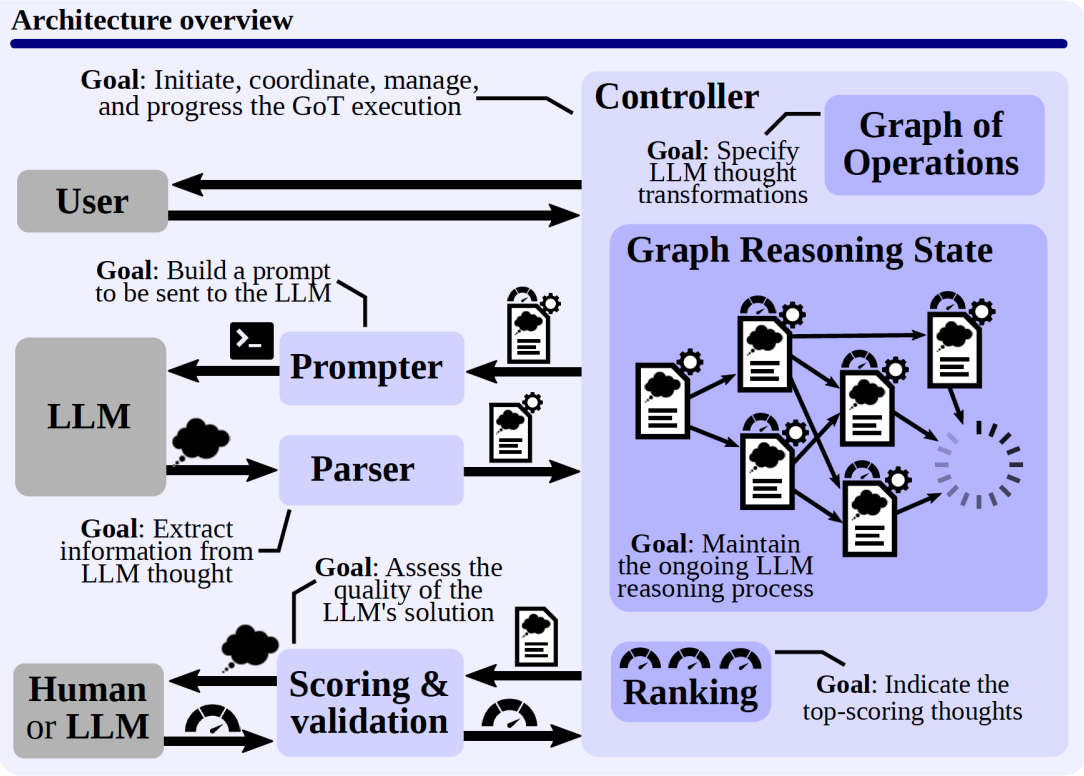


Figure 1: Comparison of Graph of Thoughts (GoT) to other prompting strategies.

Graph of Thoughts: Solving Elaborate Problems with Large Language Models. 2023.

GoT

❑ 思维图采用模块化设计，为不同任务提供了灵活的扩展策略



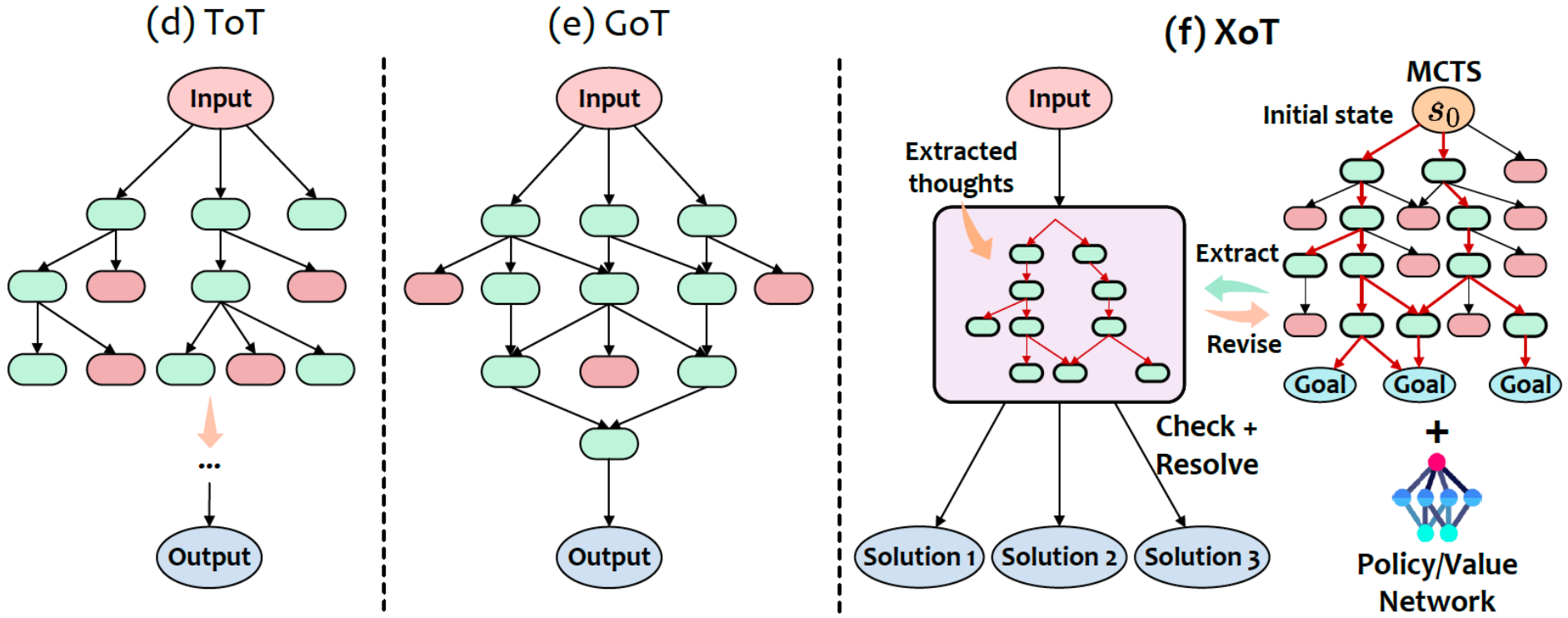
GoT包含四个核心模块：

- Prompter：为LLM准备消息
- Parser：从LLM的回复中提取信息
- Scoring：验证LLM回复并对其进行评分
- Controller：协调整个推理过程，并决定如何进行推理

Graph of Thoughts: Solving Elaborate Problems with Large Language Models. 2023.

XoT

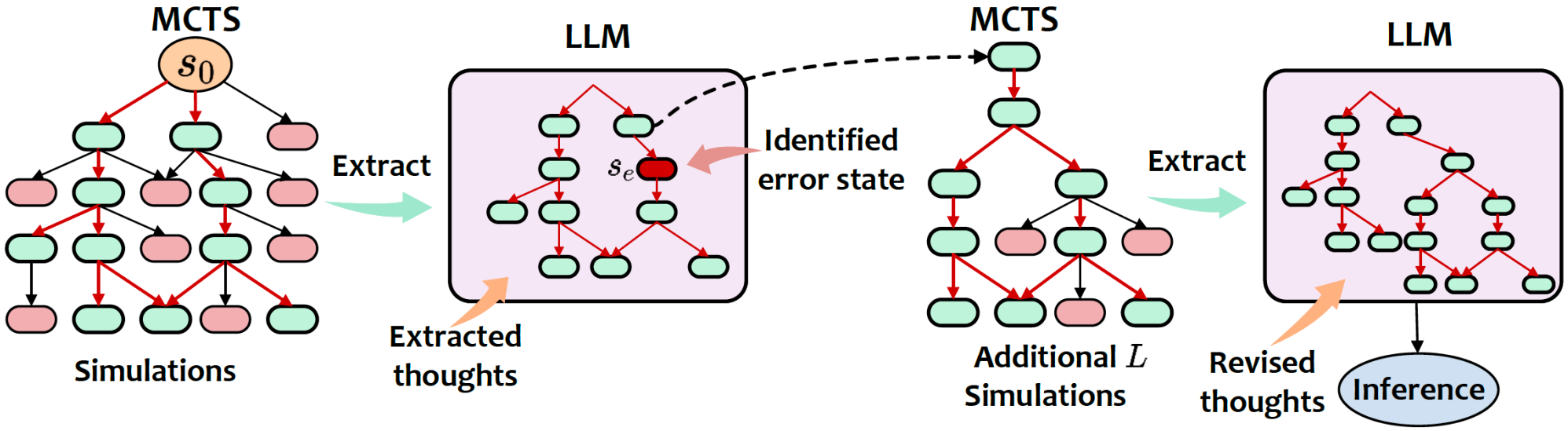
- Everything of Thoughts（XOT）将所有推理结构（链式、树状、图状）统一到一个通用框架中的推理范式



Everything of Thoughts- Defying the Law of Penrose Triangle for Thought Generation. 2023.

XoT

- XOT采用MCTS-LLM协同的思维修正机制，通过最小化LLM交互次数，自主生成高质量的认知映射



Everything of Thoughts- Defying the Law of Penrose Triangle for Thought Generation. 2023.

XoT

❑ 实验结果

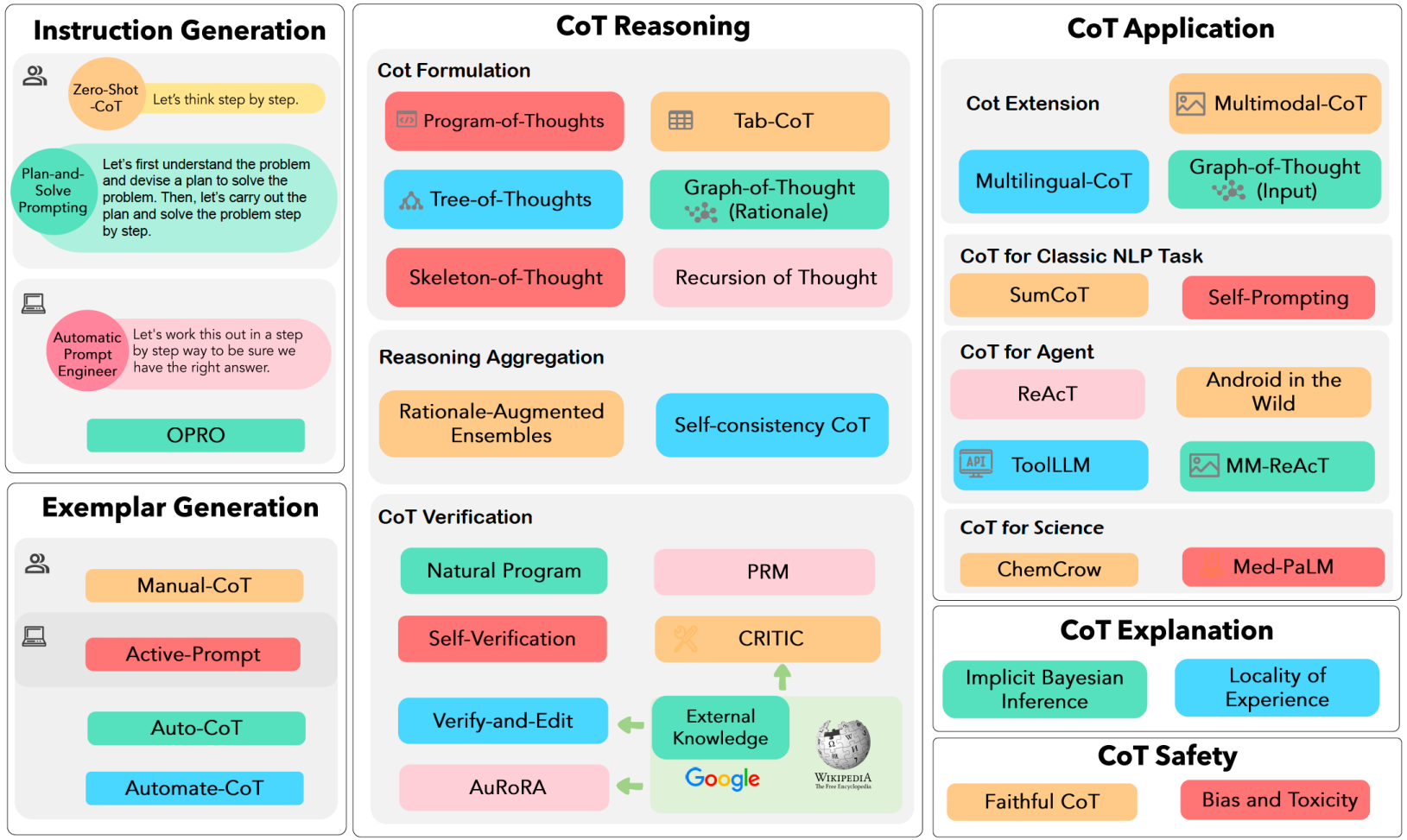
Table 3: Performance comparison on Game of 24.

Model	GPT-3.5			GPT-4		
	Acc. [%]	LLM invoked	f_θ invoked	Acc. [%]	LLM invoked	f_θ invoked
IO	6.57	1.00	-	10.22	1.00	-
CoT	2.19	1.00	-	4.38	1.00	-
CoT-SC	2.19	10.00	-	4.38	10.00	-
ToT (b=1)	5.84	22.11	-	34.31	23.50	-
ToT (b=3)	10.22	43.96	-	60.58	39.83	-
GoT (k=1)	2.92	7.00	-	10.95	7.00	-
LLaMA-2-13B	2.19	-	-	2.19	-	-
MCTS	62.77	-	-	62.77	-	-
XoT (w/ 1 r)	**79.56**	**1.39**	**92.15**	**74.45**	**1.38**	**88.20**
XoT (w/ 2 r)	**88.32**	**1.58**	**93.87**	**83.94**	**1.57**	**89.63**
XoT (w/ 3 r)	**90.51**	**1.72**	**95.94**	**85.40**	**1.78**	**92.48**

对于在 24点游戏。XoT在GPT-3.5 和 GPT-4 上均优于其他基线，修正（revision）迭代次数越多，准确率越高。

Everything of Thoughts- Defying the Law of Penrose Triangle for Thought Generation. 2023.

思维链版图



❑ CoT概览

1. 提示模式

指令生成、示例生成

2. 推理形式

CoT公式、推理聚合、CoT验证

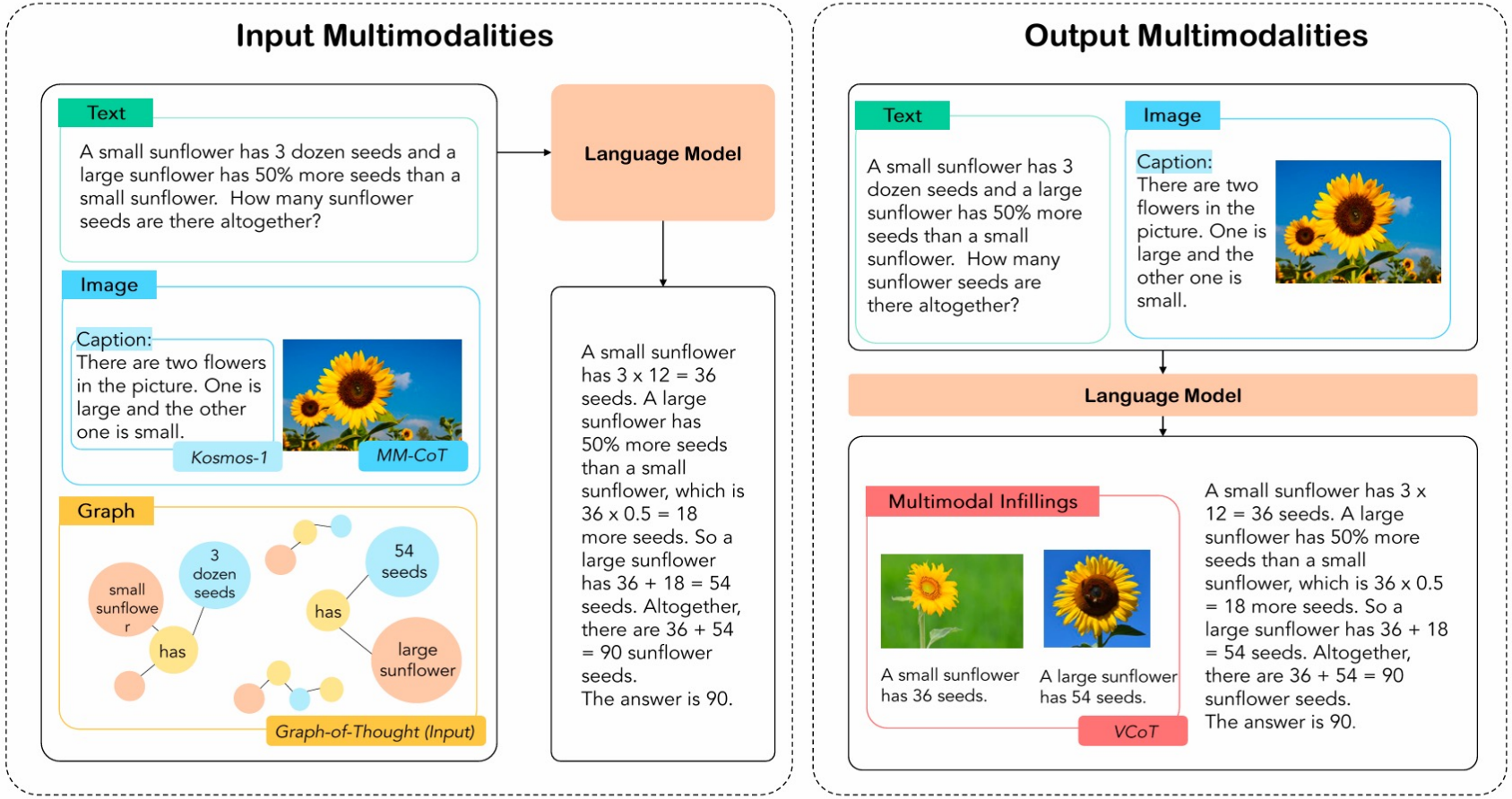
3. 应用场景

多语言、多模态、智能体以及通用任务

Igniting Language Intelligence- The Hitchhiker's Guide From Chain-of-Thought Reasoning to Language Agents. 2023.

思维链前沿应用：多模态

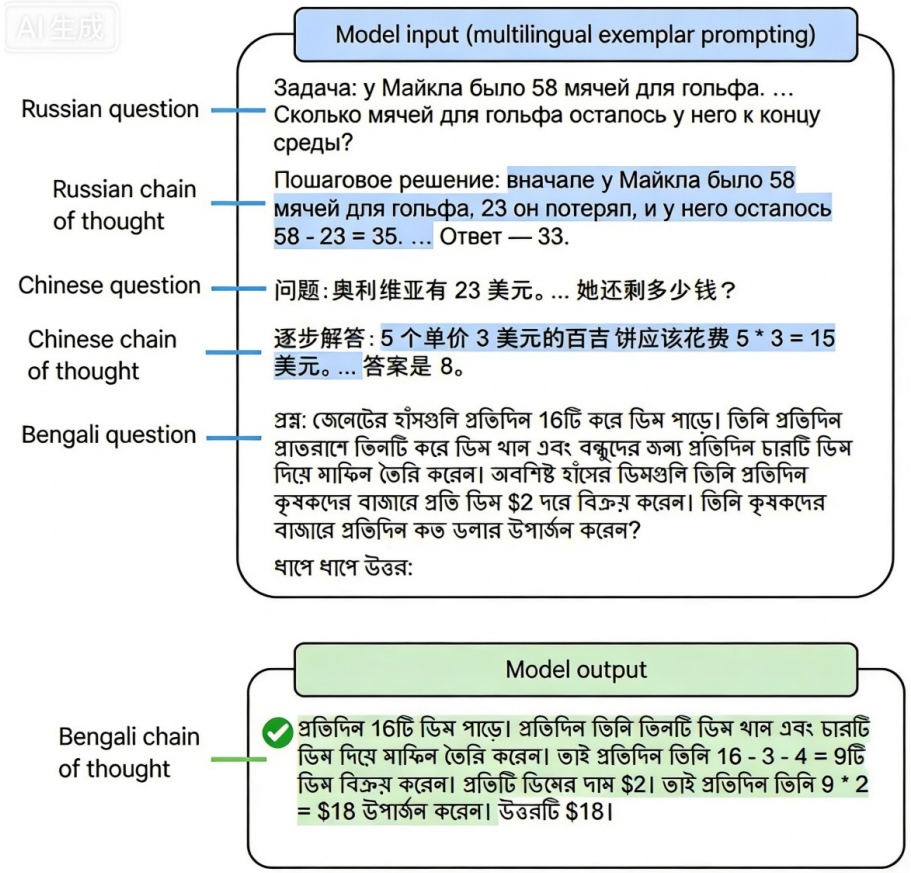
❑ MM-CoT



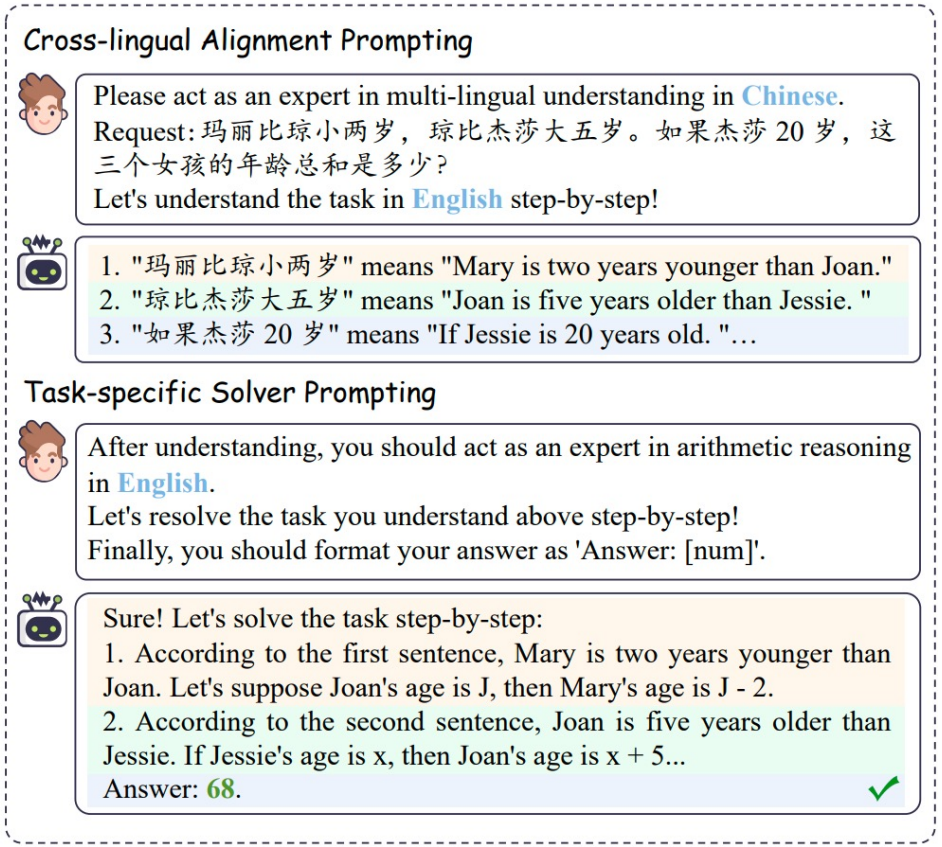
MM-CoT:A Benchmark for Probing Visual Chain-of-Thought Reasoning in Multimodal Models. 2025.

思维链前沿应用：多语言

❑ 整合多语言示例

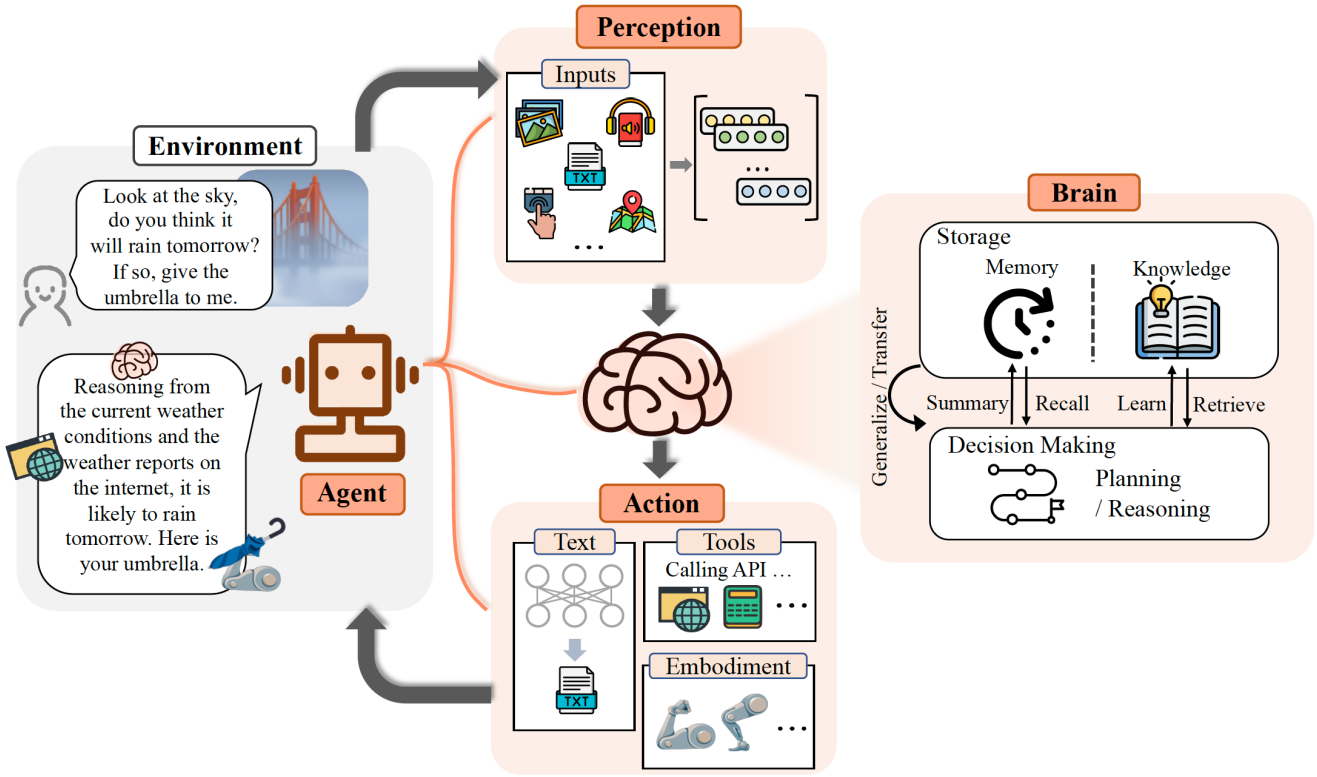


❑ 跨语言对齐提示



思维链前沿应用：智能体

- 智能体需要思维链技术来解决多步的交互式任务



(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]

The rise and potential of large language model based agents: a survey. 2023.

本节复习

- 提示学习 Prompt Learning
- 提示工程 Prompt Engineering
- 上下文学习 In-Context Learning
- 思维链 Chain of Thought

参考文献

- Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. 2021.
- Language Models are Few-Shot Learners. 2020.
- A Survey on In-context Learning. 2022.
- An Explanation of In-context Learning as Implicit Bayesian Inference. 2021.
- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. 2022.

致 谢

- 胡玥、曹亚男、方芳：国科大《自然语言处理基础》
- 曹亚男、任昱冰：国科大《深度学习与自然语言处理概述》


THANKS
https://ictkc.github.io/teaching/2026spring-nlp