Prompt-based Learning

Tanmoy Chakraborty
Associate Professor, IIT Delhi
https://tanmoychak.com/

Many slides from Mohit Iyyer, Graham Neubig

Recommended Reading

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

Pengfei Liu
Carnegie Mellon University
pliu3@cs.cmu.edu

Weizhe Yuan
Carnegie Mellon University
weizhey@cs.cmu.edu

Jinlan Fu
National University of Singapore
jinlanjonna@gmail.com

Zhengbao Jiang
Carnegie Mellon University
zhengbaj@cs.cmu.edu

Hiroaki Hayashi
Carnegie Mellon University
hiroakih@cs.cmu.edu

Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu

The Language Model “Scaling Wars”!

ELMo: 93M params, 2-layer biLSTM

BERT-base: 110M params, 12-layer Transformer

BERT-large: 340M params, 24-layer Transformer

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

The Language Model “Scaling Wars”!

ELMo: 93M params, 2-layer biLSTM

BERT-base: 110M params, 12-layer Transformer

BERT-large: 340M params, 24-layer Transformer

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

The Language Model "Scaling Wars"!

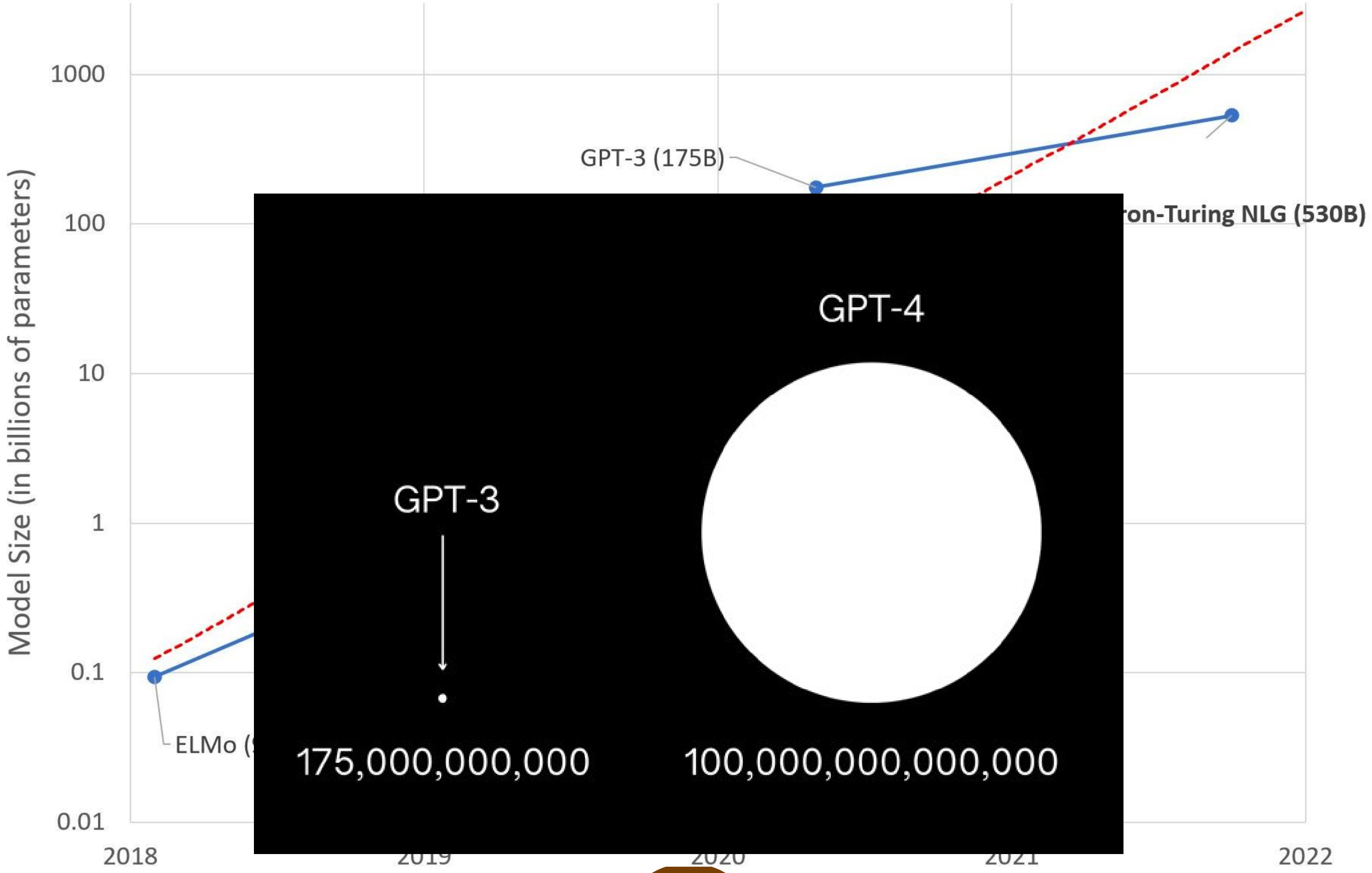
ELMo: 1B training tokens

BERT: 3.3B training tokens

RoBERTa: ~30B training tokens

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Colossal Models



So… What Does All of This Scaling Buy Us?

GPT-3

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

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

Downstream training data

Downstream test data

Zero-shot

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

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

Zero-shot

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

```
Translate English to French:
cheese =>
```

task description

prompt

No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:

We will see how LLMs are very 'sensitive' to such prompt formatting, and how we can measure this sensitivity!

"Translate English to French: cheese =>"

Why "=>" ? What is the optimal prompt?

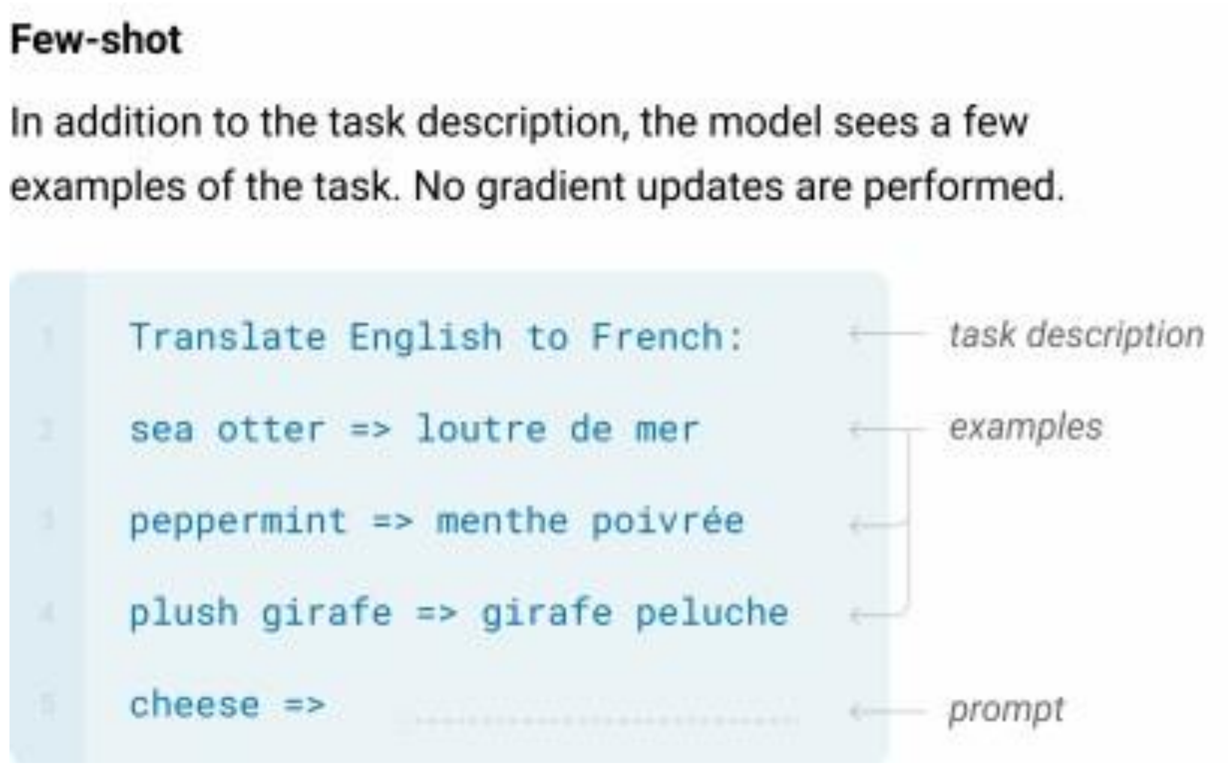
One-shot

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

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

No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:

"Translate English to French: sea otter => loutre de mer, cheese =>"



No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:

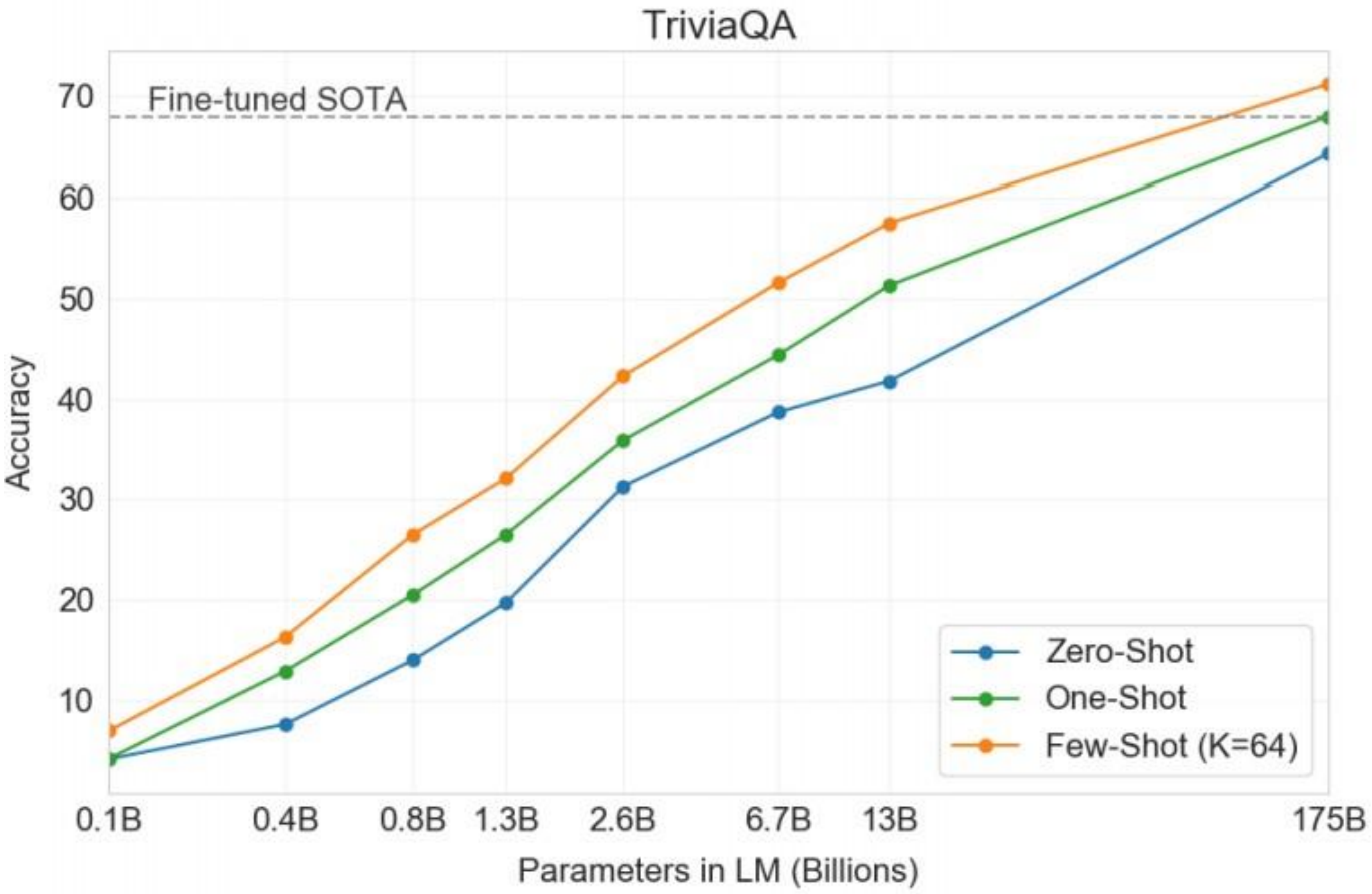
Many such examples fed into the prefix in this way

“Translate English to French: sea otter => loutre de mer, peppermint => … (few more examples) , cheese => ”

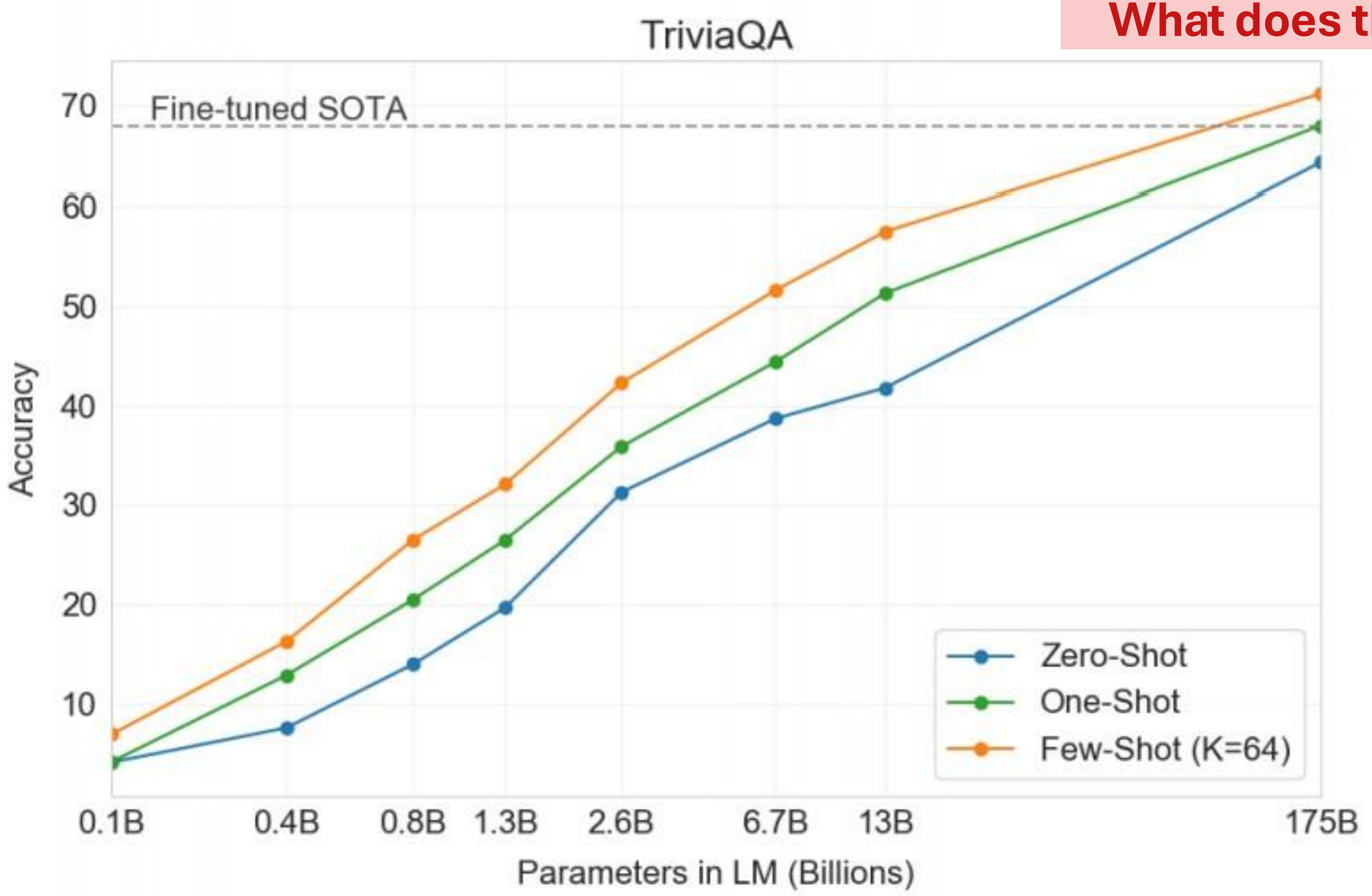
How Does This New Paradigm Compare to “Pretrain + Finetune”?

TriviaQA

Question
Miami Beach in Florida borders which ocean?
What was the occupation of Lovely Rita according to the song by the Beatles
Who was Poopdeck Pappys most famous son?
The Nazi regime was Germany's Third Reich; which was the first Reich?
At which English racecourse did two horses collapse and die in the parade ring due to electrocution, in February 2011?
Which type of hat takes its name from an 1894 novel by George Du Maurier where the title character has the surname O'Ferrall ?
What was the Elephant Man's real name?


TriviaQA
Fine-tuned SOTA
70
60
50
40
30
20
10
Accuracy
0.1B
0.4B
0.8B
1.3B
2.6B
6.7B
13B
175B
Parameters in LM (Billions)
Zero-Shot
One-Shot
Few-Shot (K=64)

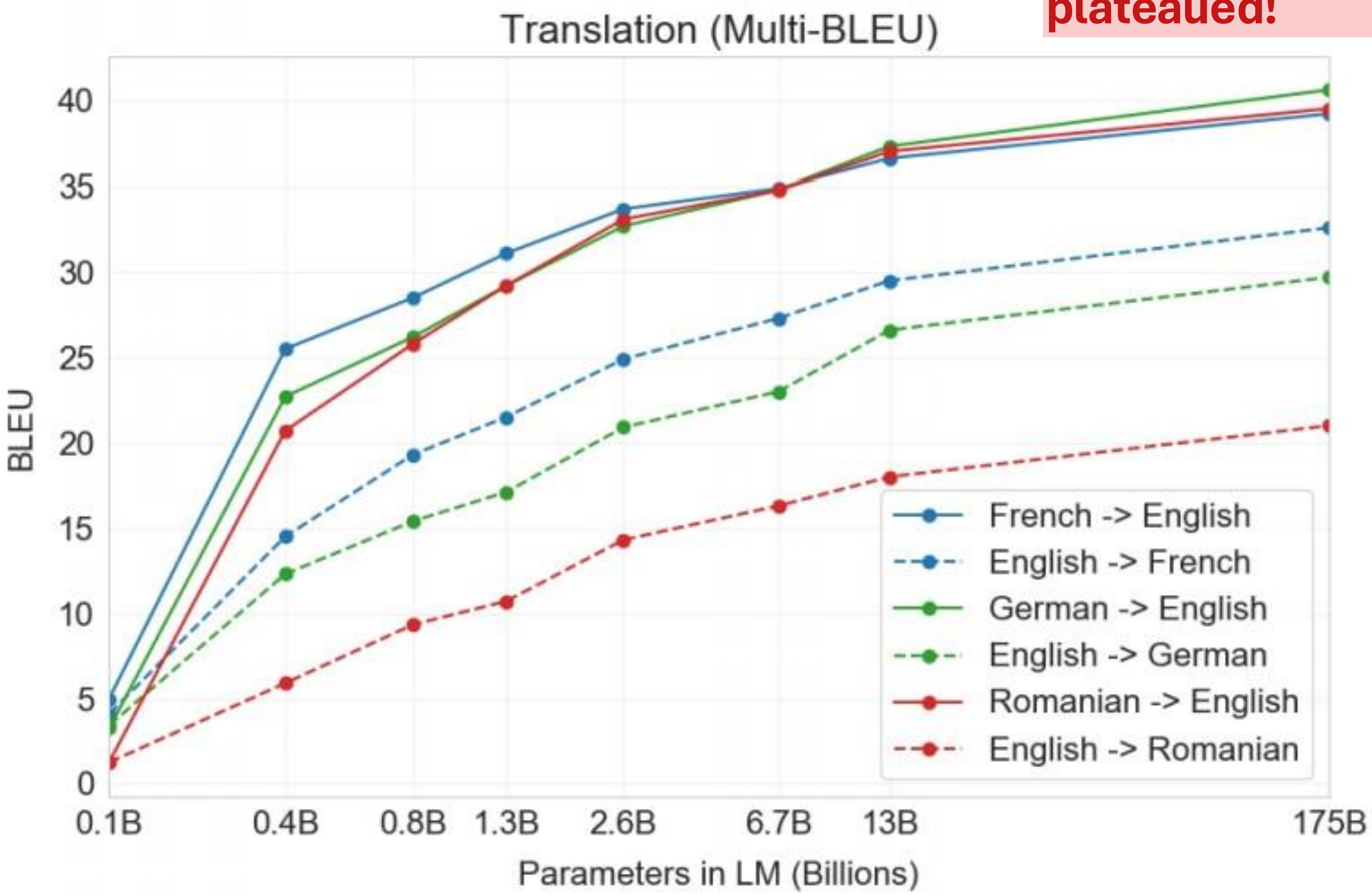
What does this mean?



What About Translation?
(7% of GPT3's Training Data is in Languages Other Than English)

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	<u>37.5</u>	34.9	28.3	35.2	<u>35.2</u>	33.1
mBART [LGG+20]	-	-	<u>29.8</u>	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	<u>39.2</u>	29.7	<u>40.6</u>	21.0	<u>39.5</u>

Improvements haven't plateaued!

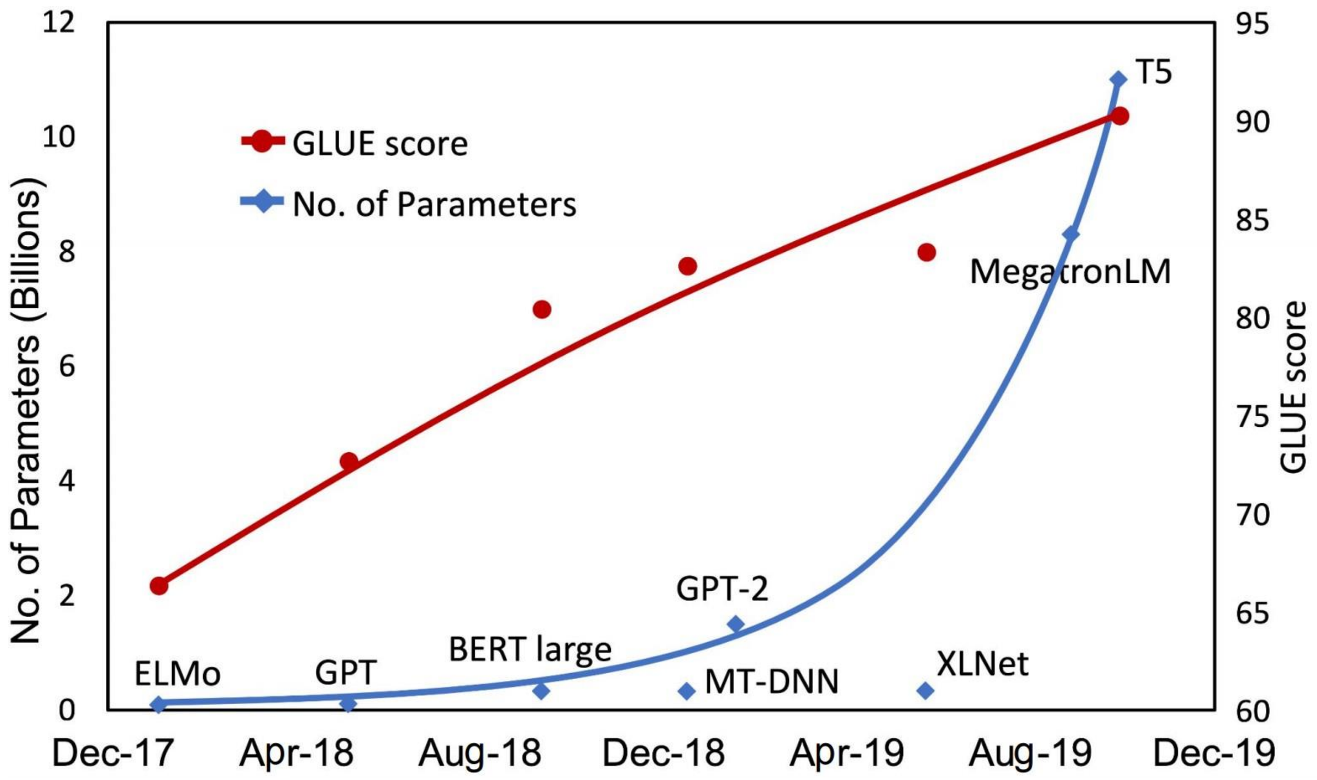


What About Reading Comprehension QA?

Setting	CoQA	DROP	QuAC	SQuADv2	RACE-h	RACE-m
Fine-tuned SOTA	**90.7**[a]	**89.1**[b]	**74.4**[c]	**93.0**[d]	**90.0**[e]	**93.1**[e]
GPT-3 Zero-Shot	81.5	23.6	41.5	59.5	45.5	58.4
GPT-3 One-Shot	84.0	34.3	43.3	65.4	45.9	57.4
GPT-3 Few-Shot	85.0	36.5	44.3	69.8	46.8	58.1

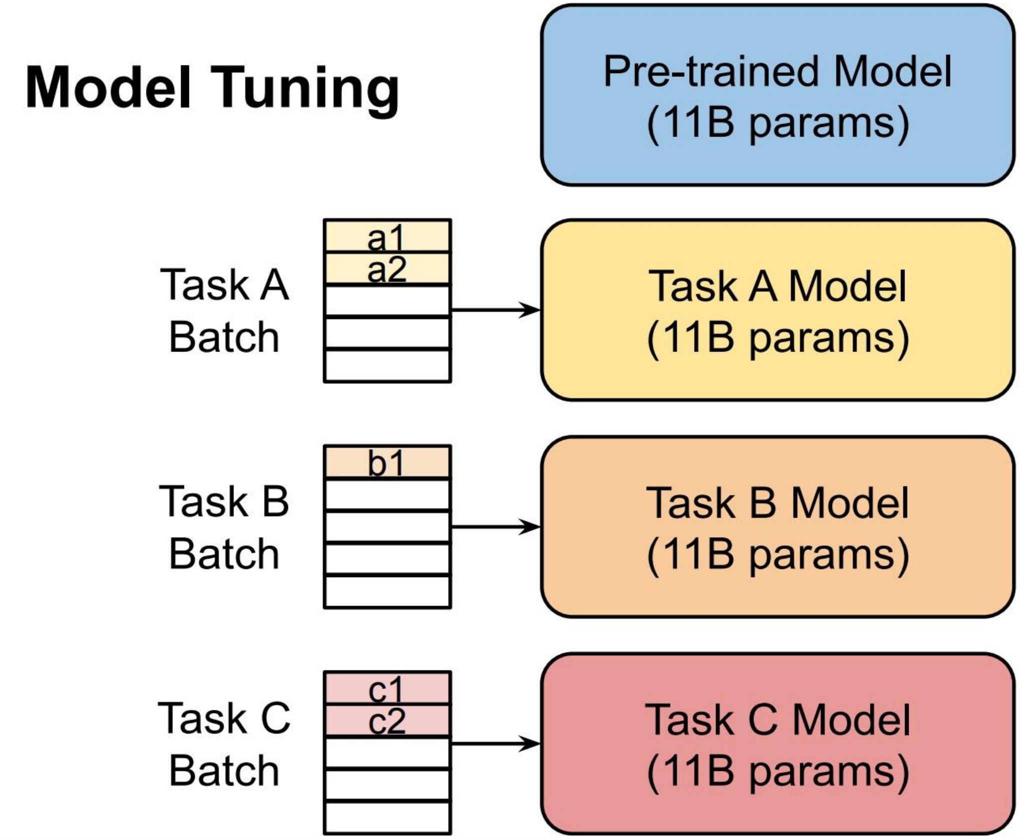
Struggles on "harder" datasets

Scaling up the model size is one of the most important ingredients for achieving the best performance



[Ahmet and Abdullah., 2021](#)

Practical Challenges: Large-Scale Models are Costly to Share and Serve



[Lester et al., 2021](#)

Language Model Prompting to The Rescue!

GPT-3 (Brown et al., 2020): **In-context learning**

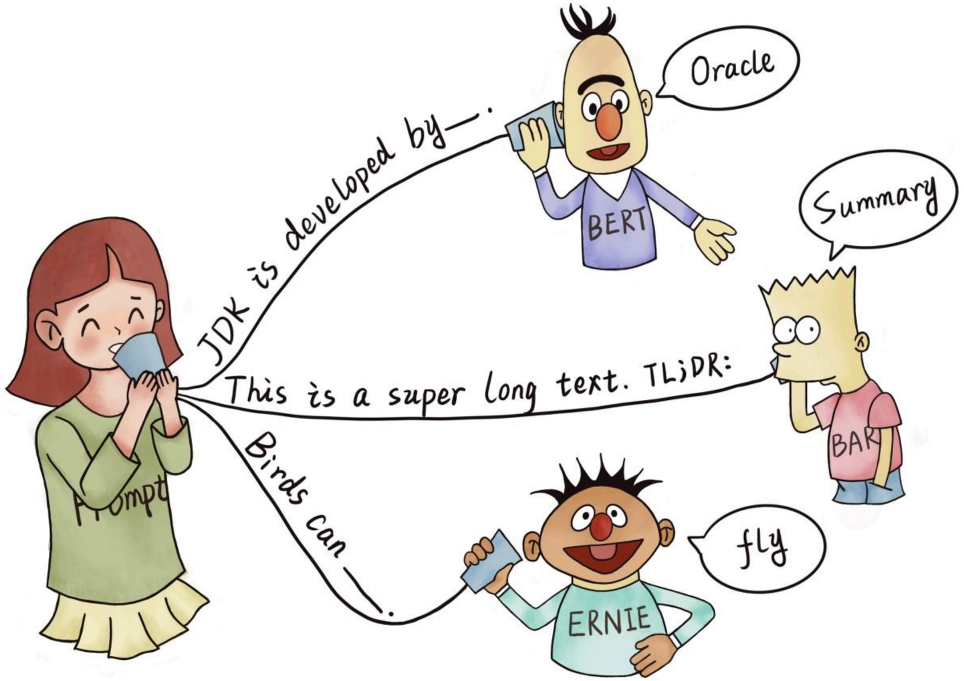
- **natural language instruction** and/or **a few task demonstrations** ⟶ **output**

  "Translate English to German:" That is good ⟶ Das is gut

- *no* gradient updates or fine-tuning

What is Prompting？

Encouraging a pre-trained model to make particular predictions by providing a "prompt" specifying the task to be done.



Terminologies and Notations

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]`

Terminology and notation of prompting methods. $\boldsymbol{z}^*$ represents answers that correspond to true output $\boldsymbol{y}^*$.

What’s The General Workflow of Prompting?

- Prompt Addition
- Answer Prediction
- Answer-Label Mapping

Prompt Addition

Prompt Addition: Given input x, we transform it into prompt x' through two steps:

1. Define a template with two slots, one for input [x], and one for the answer [z]
2. Fill in the input slot [x]

Example: Sentiment Classification

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."

Answer Prediction

Answer Prediction: Given a prompt, predict the answer [z]

- Fill in [z]

Example

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."

⇩

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

Mapping

- **Mapping:** Given an answer, map it into a class label

Example

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.”

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

Mapping: fantastic => Positive

Types of Prompts

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

  - Filled Prompt: **I love this movie. Overall it was a boring movie**
  - Answered Prompt: **I love this movie. Overall it was a fantastic movie**

  - Prefix Prompt: **I love this movie. Overall this movie is [z]**
  - Cloze Prompt: **I love this movie. Overall it was a [z] movie**

Sub-optimal and Sensitive Discrete/Hard Prompts

- **Discrete/hard prompts**
  - natural language instructions/task descriptions
- **Problems**
  - requiring domain expertise/understanding of the model's inner workings
  - performance still lags far behind SoTA model tuning results
  - sub-optimal and sensitive
    - prompts that humans consider reasonable is not necessarily effective for language models
    - pre-trained language models are sensitive to the choice of prompts

Sub-optimal and Sensitive Discrete/Hard Prompts

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

Table 1. Case study on LAMA-TREx P17 with bert-base-cased. A single-word change in prompts could yield a drastic difference.

Liu et al., 2021

Shifting From Discrete/Hard to Continuous/Soft Prompts

Progress in prompt-based learning

- manual prompt design (Brown et al., 2020; Schick and Schutze, 2021a,b)
- mining and paraphrasing based methods to automatically augment the prompt sets (Jiang et al., 2020)
- gradient-based search for improved discrete/hard prompts (Shin et al., 2020)
- automatic prompt generation using a separate generative language model (i.e., T5) (Gao et al., 2020)
- learning continuous/soft prompts (Liu et al., 2021; Li and Liang., 2021; Qin and Eisner., 2021; Lester et al., 2021)

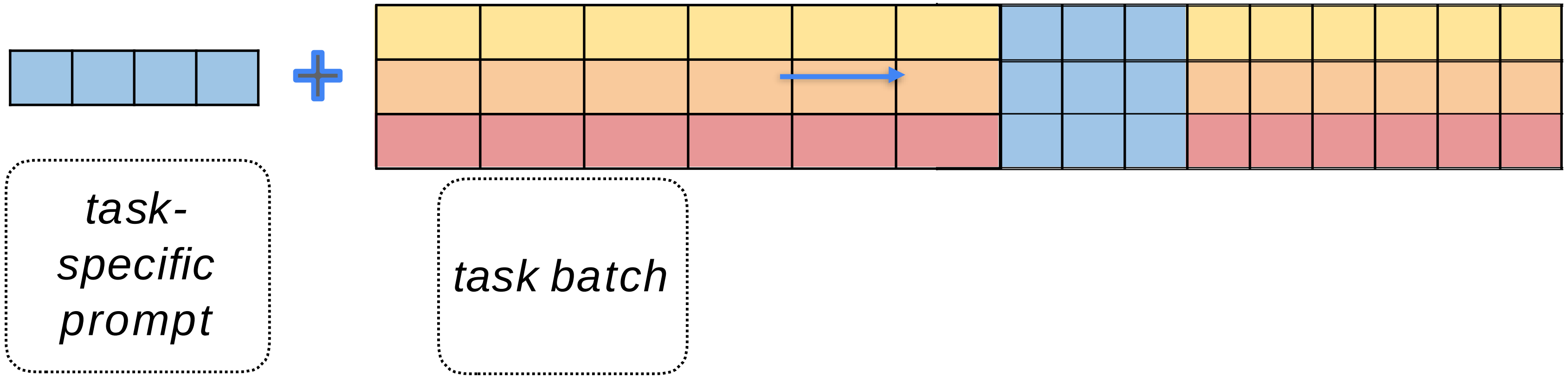
Continuous/soft prompts

- additional learnable parameters injected into the model

Prompt Tuning Idea

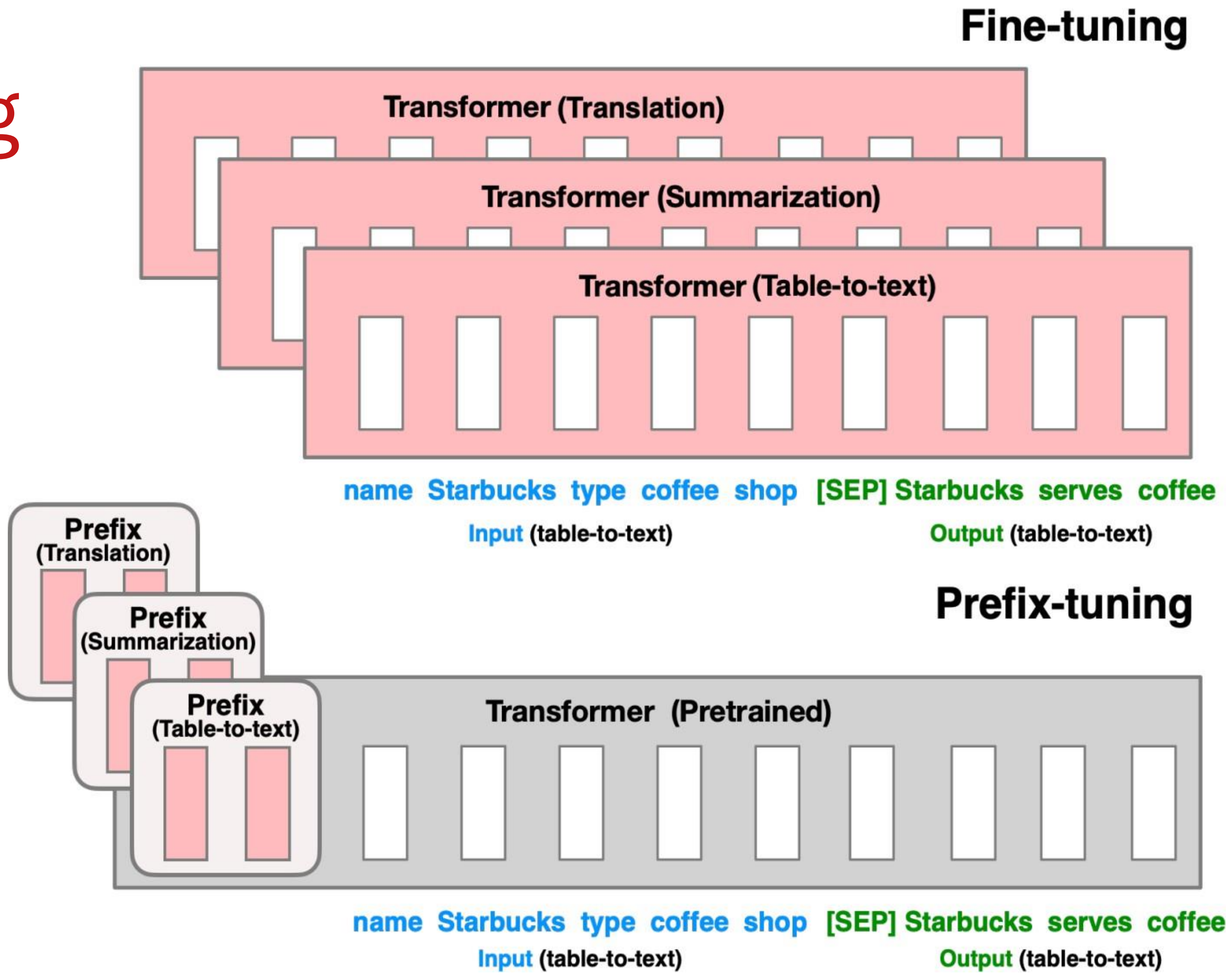
What is a prompt in Prompt Tuning?

A sequence of additional task-specific tunable tokens prepended to the input text



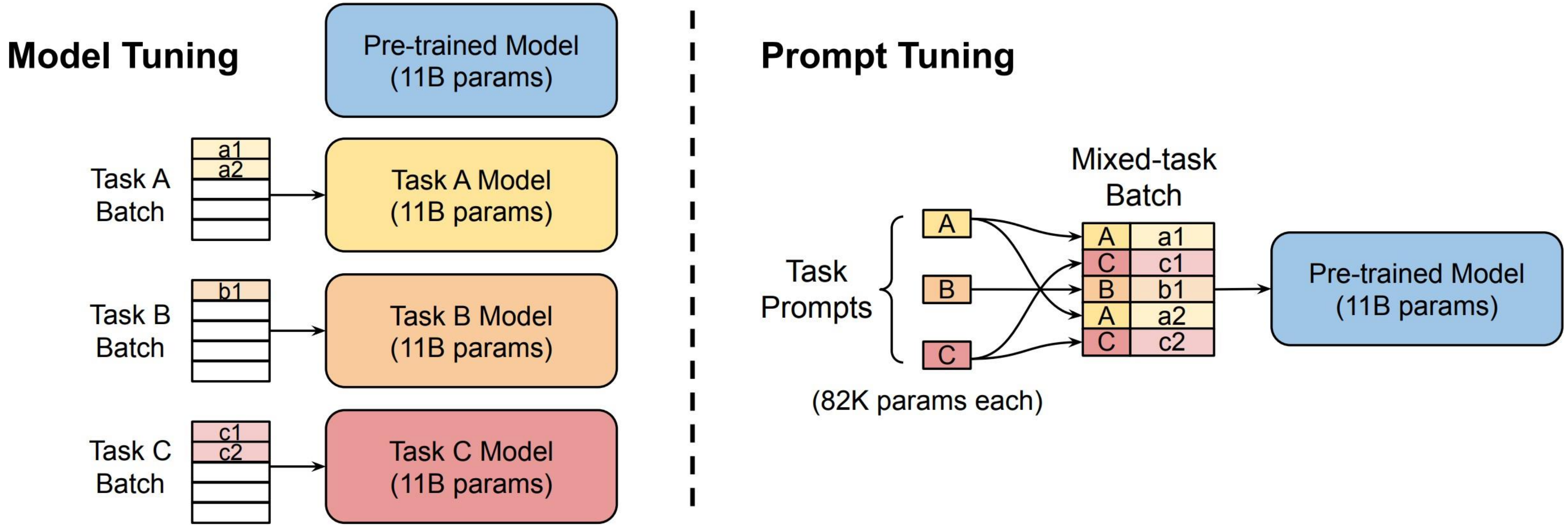
(Lester et al., 2021)

Prefix Tuning

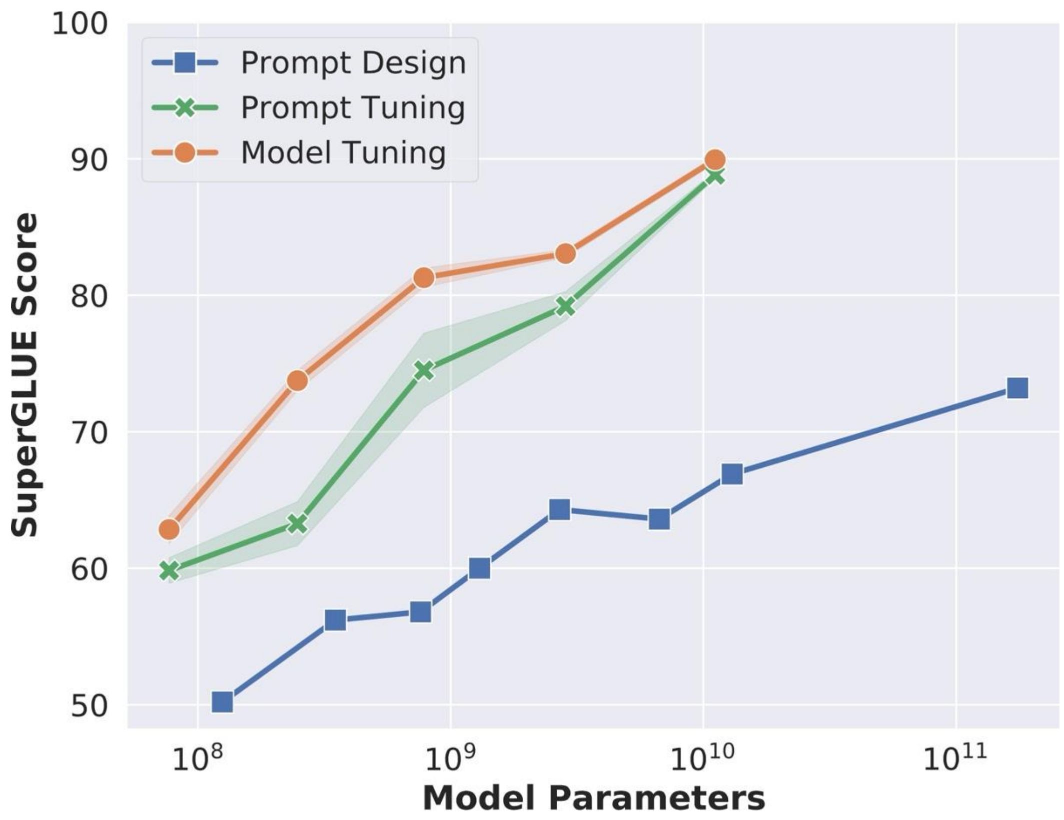


Li & Liang, ACL 2021

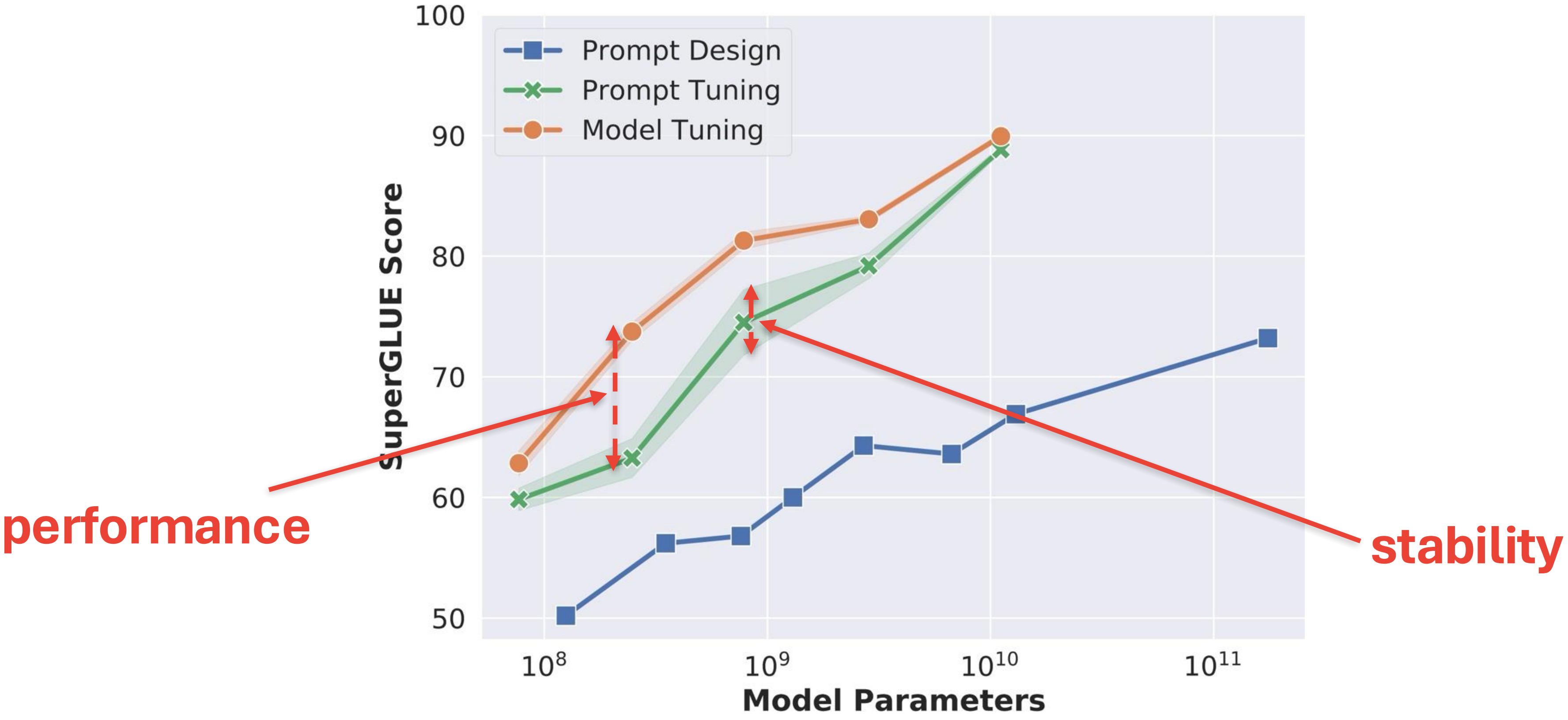
Parameter-efficient Prompt Tuning



Prompt Tuning Becomes More Competitive With Scale

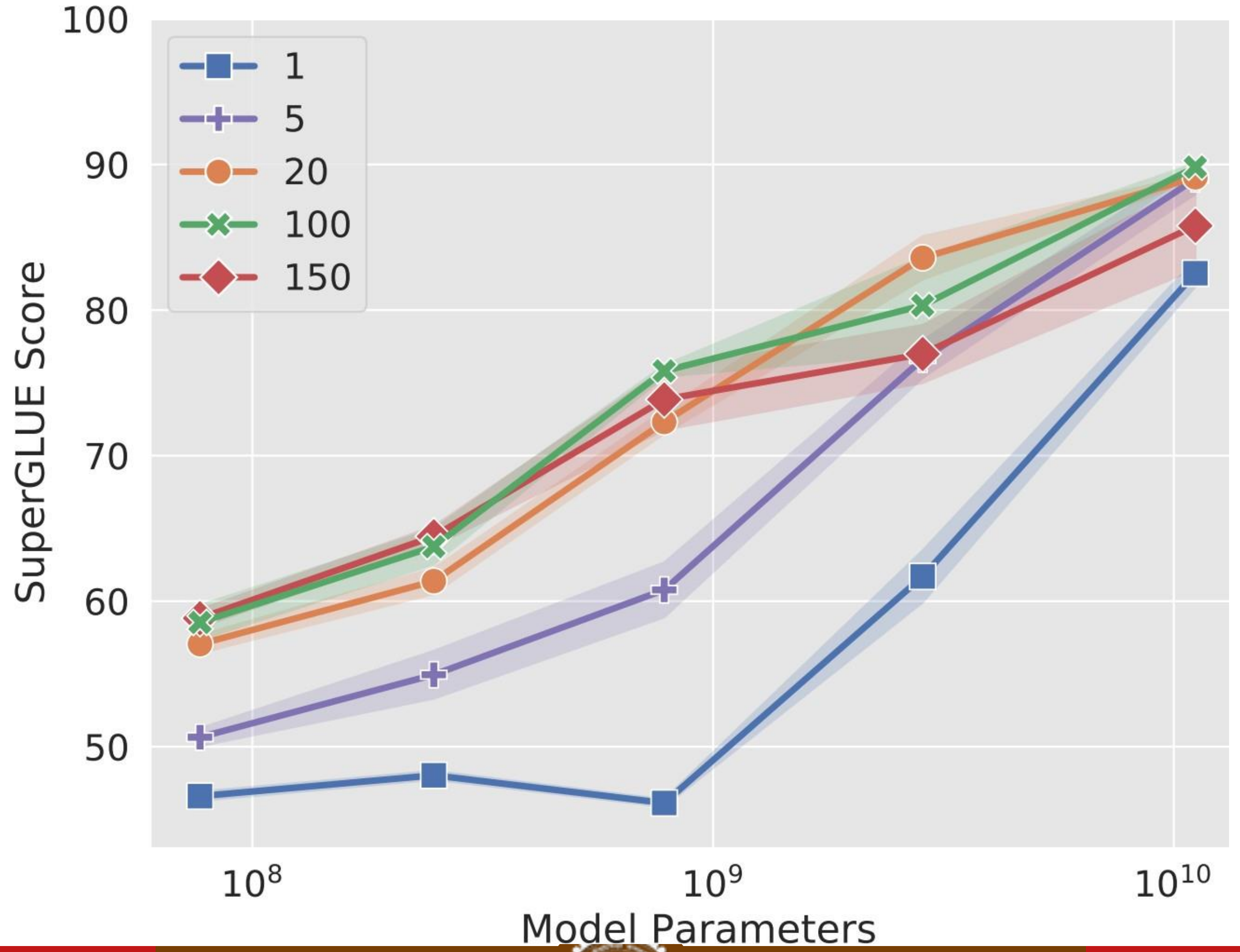


Room for Improving Prompt Tuning

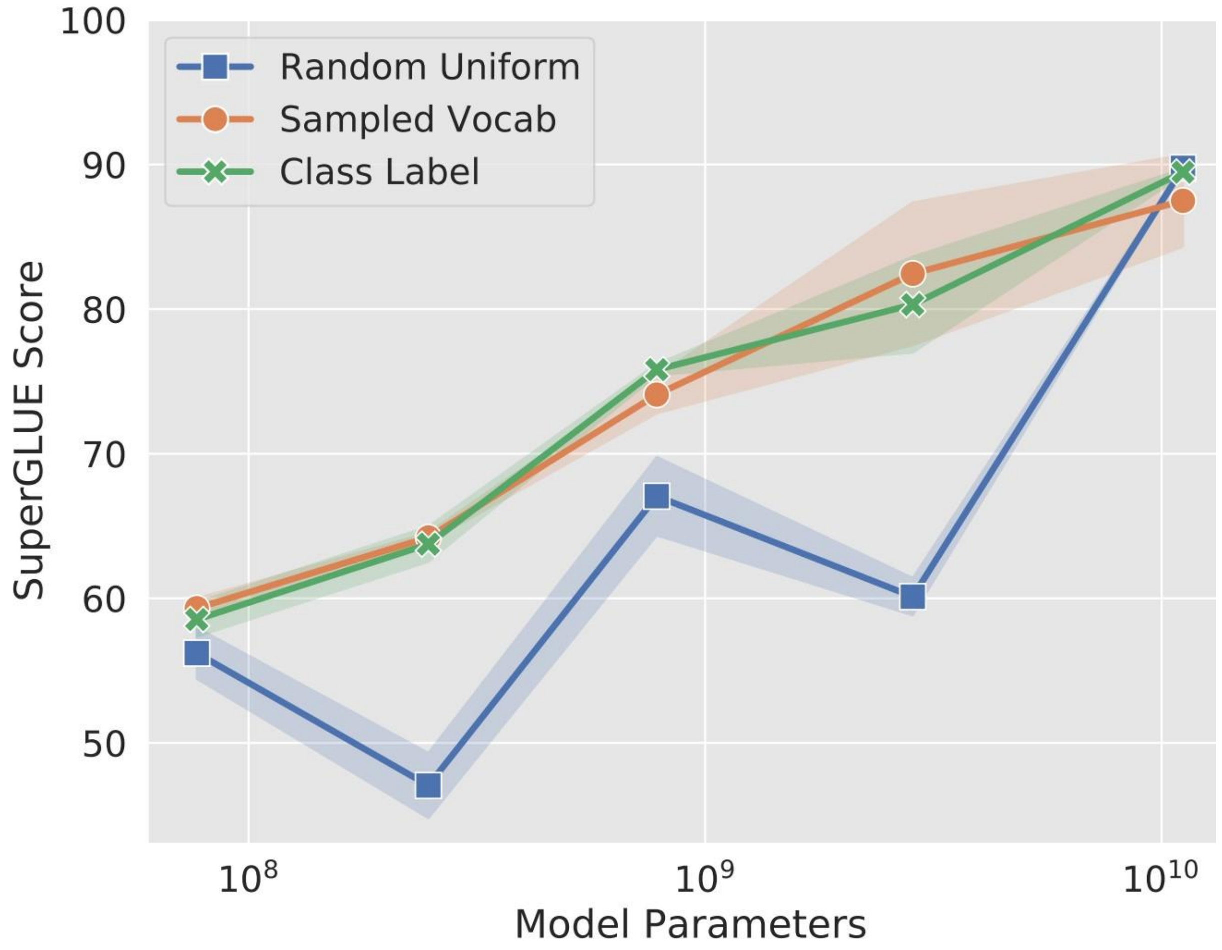


Lester et al., 2021

Prompt Length Matters Less With Larger Pre-trained LMs



Prompt Initialization Matters Less With Larger Pre-trained LMs



Problems With Soft Prompts

- Requires separate training
- Not possible to get soft prompts for all possible tasks and inputs
- Not user-friendly
  - How will non-expert users get soft prompts for new tasks/inputs while interacting with the LMs?

Hard prompts, thus, continue to be the default choice for interacting/utilizing LLMs.

Advanced Prompting

Prompting vs CoT

Standard Prompting

Model Input

Q: Mohit 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: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and brought 6 more how many apples do they have?

Model Output

A: The answer is 27.

Chain-of-Thought Prompting

Model Input

Q: Mohit 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: Mohit started with 5 balls. 2 cans of 3 tennis balls 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and brought 6 more how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

Prompting vs CoT

Chain-of-Thought Prompting

Standard Prompting

Model Input

Q: Mohit 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: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and brought 6 more how many apples do they have?

Model Output

A: The answer is 27.

Model Input

Q: Mohit 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?

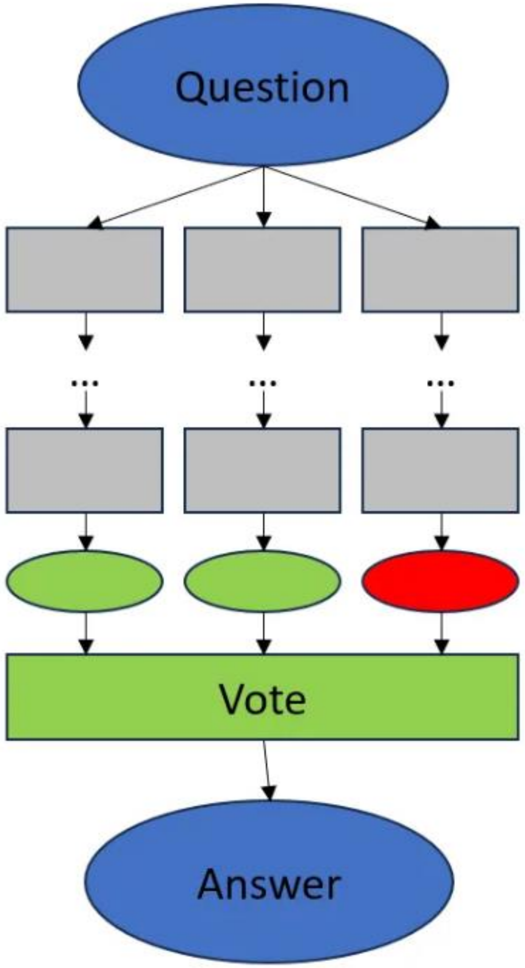
A: Mohit started with 5 balls. 2 cans of 3 tennis balls 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and brought 6 more how many apples do they have?

Model Output

A: The cafeteria had 23 appl iginally. They used20 t ke lunch. S y had 23 - 20 = 3. They bou 6 mor les, so they have 3 + 6 = 9. The an

CoT with Self Consistency



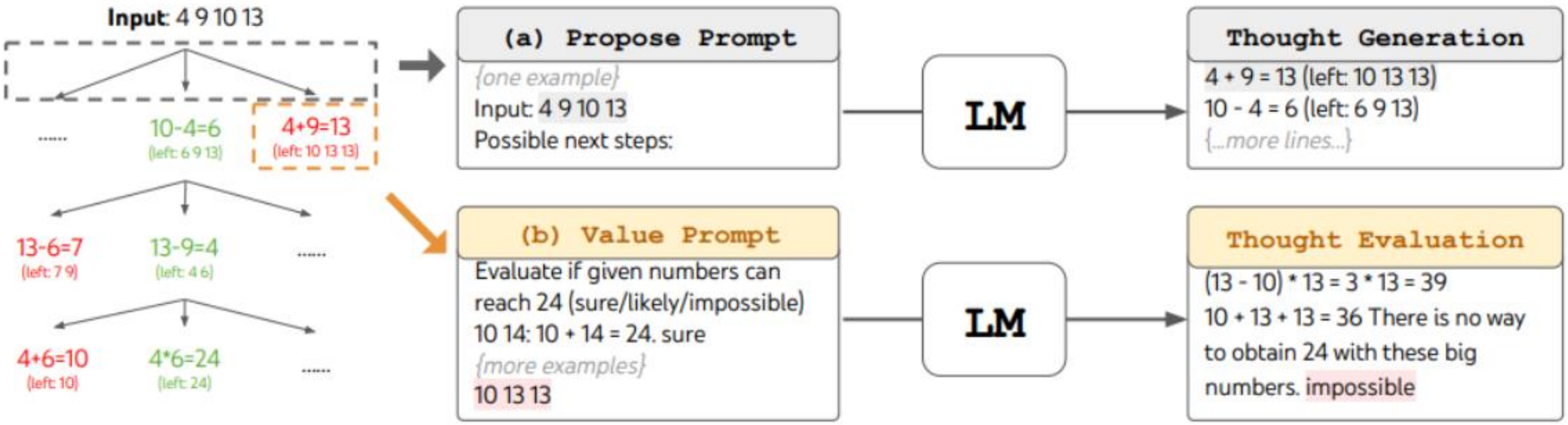
Procedure

1. Add „think step-by-step" to your original question (we'll call this augmented question the *question* in the following).
2. Ask the question repeatedly (n times) and collect the answers.
3. Decide for a voting technique and decide which of the collected answers is picked as the final answer.

https://medium.com/@johannes.koeppern/self-consistency-with-chain-of-thought-cot-sc-2f7a1ea9f941

Tree-of-Thought (ToT)

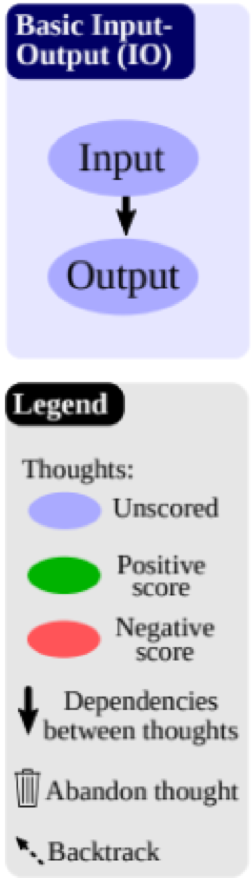
- **Key components:**
  - **Branching:** Generates multiple thought paths for each step
  - **Scoring:** Evaluates quality of each thought/path
  - **Backtracking:** Returns to previous points if a path is unproductive

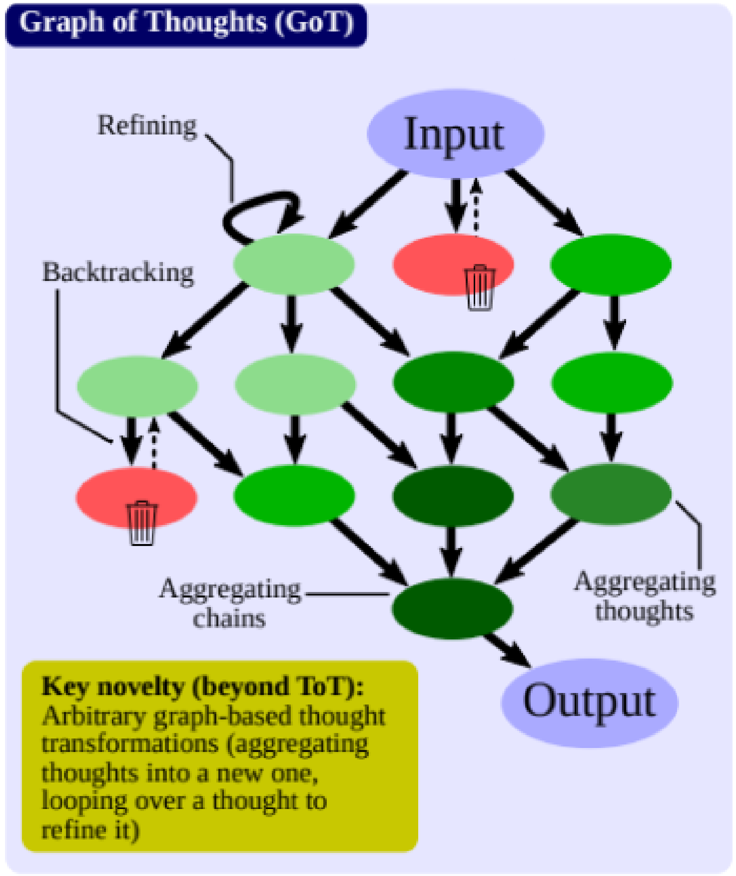


https://wandb.ai/sauravmaheshkar/prompting-techniques/reports/Chain-of-thought-tree-of-thought-and-graph-of-thought-Prompting-techniques-explained---Vmlldzo4MzQwNjMx

Graph-of-Thought (GoT)

- **Refining:** Modifies existing thoughts by adding loops in the graph
- **Aggregating:** Combines multiple thoughts into new ones by creating vertices with multiple incoming edges

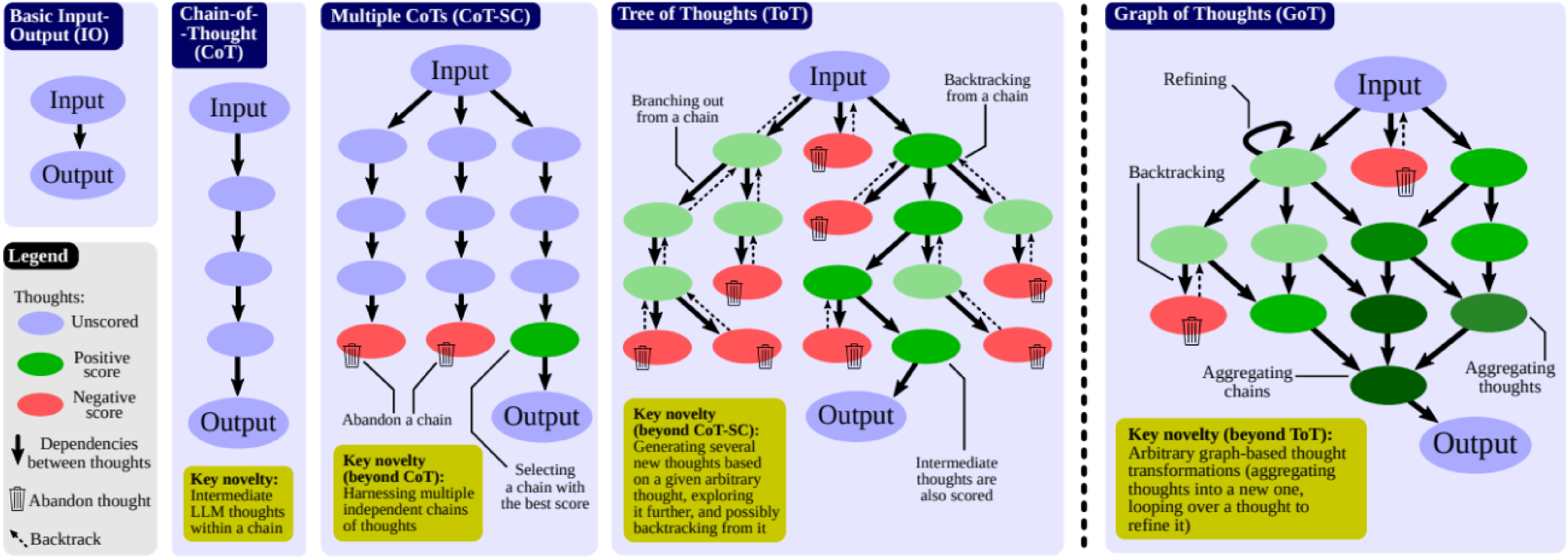




https://wandb.ai/sauravmaheshkar/prompting-techniques/reports/Chain-of-thought-tree-of-thought-and-graph-of-thought-Prompting-techniques-explained---Vmlldzo4MzQwNjMx

Graph-of-Thought (GoT)

- **Refining:** Modifies existing thoughts by adding loops in the graph
- **Aggregating:** Combines multiple thoughts into new ones by creating vertices with multiple incoming edges



https://wandb.ai/sauravmaheshkar/prompting-techniques/reports/Chain-of-thought-tree-of-thought-and-graph-of-thought-Prompting-techniques-explained---Vmlldzo4MzQwNjMx

However, LMs Continue to be Sensitive to Minor Prompt Variations

Small Changes in Prompts Can Lead to Big 'Surprises'!

Meta Llama 3
8B Instruct

Q: How much are you familiar with the principles of Buddhism?\nA:

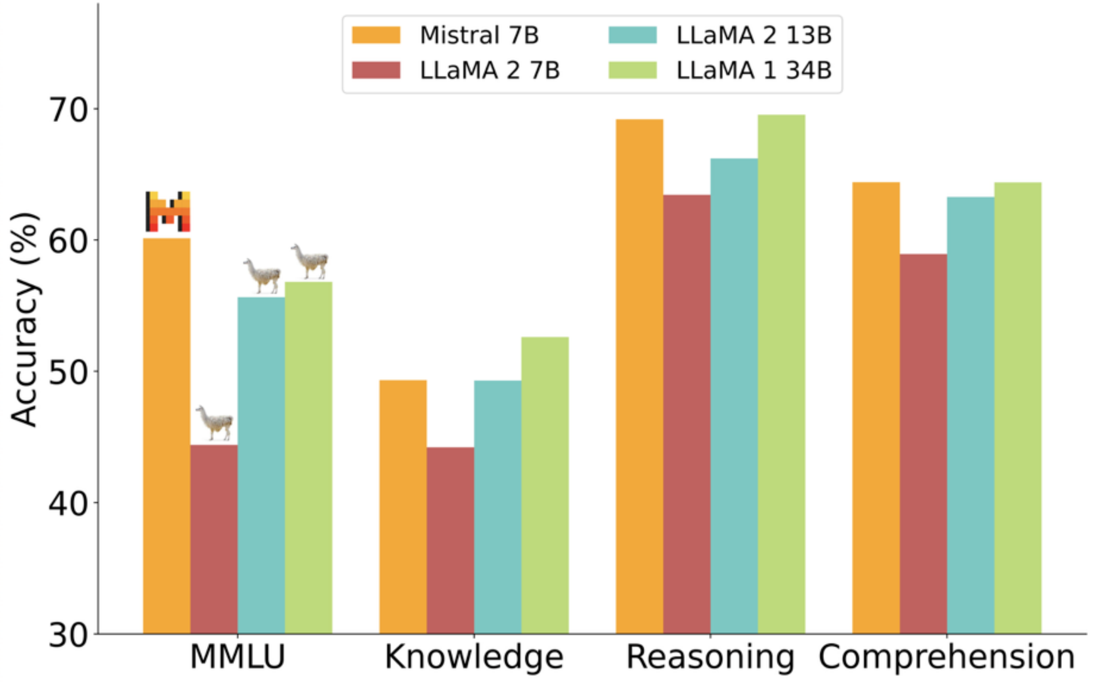
Buddhism is a philosophy and spiritual practice that originated in ancient India ...

Q: How much do you understand Buddhism?\nA:

0.000001% (just kidding, but I'm not a Buddhist scholar either!)

Is Accuracy Enough?

	Meta Llama 3 8B	Gemma 7B - It Measured	Mistral 7B Instruct Measured
MMLU 5-shot	68.4	53.3	58.4
GPQA 0-shot	34.2	21.4	26.3
HumanEval 0-shot	62.2	30.5	36.6
GSM-8K 8-shot, CoT	79.6	30.6	39.9
MATH 4-shot, CoT	30.0	12.2	11.0



- Only Accuracy (or, a measure of correctness) reported.
- None of the models report prompt sensitivity on benchmarks!
- **No standard measure for capturing prompt sensitivity exists !!!**

Sensitivity is Orthogonal to Correctness

Model-A

Performance on a benchmark of interest	Prompt Sensitivity
0.85	0.6

Model-B

Performance on a benchmark of interest	Prompt Sensitivity
0.75	0.2

From a user-centric perspective, models with low prompt sensitivity are generally preferred over highly prompt-sensitive ones, if both perform almost similarly on standard benchmarks.

Thus, **Model-B** is often **preferred** by a user **over** **Model-A**.

How to Measure Sensitivity to Prompts?

Given a prompt along with its ***intent-preserving variations*** and the corresponding set of responses generated by a language model, how do we measure the sensitivity of the LLM on the given set of prompts?

The measure should work for:

- All variation types
- All task types (open-ended generation & MCQs/classification tasks)

POSIX: A Novel PrOmpt Sensitivity IndeX

POSIX

A Prompt Sensitivity Index for Language Models

```
pip install prompt-sensitivity-index
```

POSIX: A Novel PrOmpt Sensitivity IndeX

POSIX: A Prompt Sensitivity Index For Large Language Models

Anwoy Chatterjee[*†]
Dept. of Electrical Engineering
Indian Institute of Technology Delhi
anwoy.chatterjee@ee.iitd.ac.in

H S V N S Kowndinya Renduchintala[†]
Media and Data Science Research
Adobe Inc., India
rharisrikowndinya333@gmail.com

Sumit Bhatia
Media and Data Science Research
Adobe Inc., India
sumit.bhatia@adobe.com

Tanmoy Chakraborty
Dept. of Electrical Engineering
Indian Institute of Technology Delhi
tanchak@iitd.ac.in

EMNLP-findings’24

Intent-preserving or *Intent-aligned* Prompt Variations

What is the capital of India?

- What city serves as the capital of India?
- Can you tell me the capital city of India?
- Where is the capital of India located?
- What is the name of India's capital?
- Can you provide the name of India's capital?

Intent-preserving / Intent-aligned variants

- User expects same response

What Aspects Should be Captured?

1. Response Diversity
2. Response Distribution Entropy
3. Semantic Coherence
4. Variance in Confidence

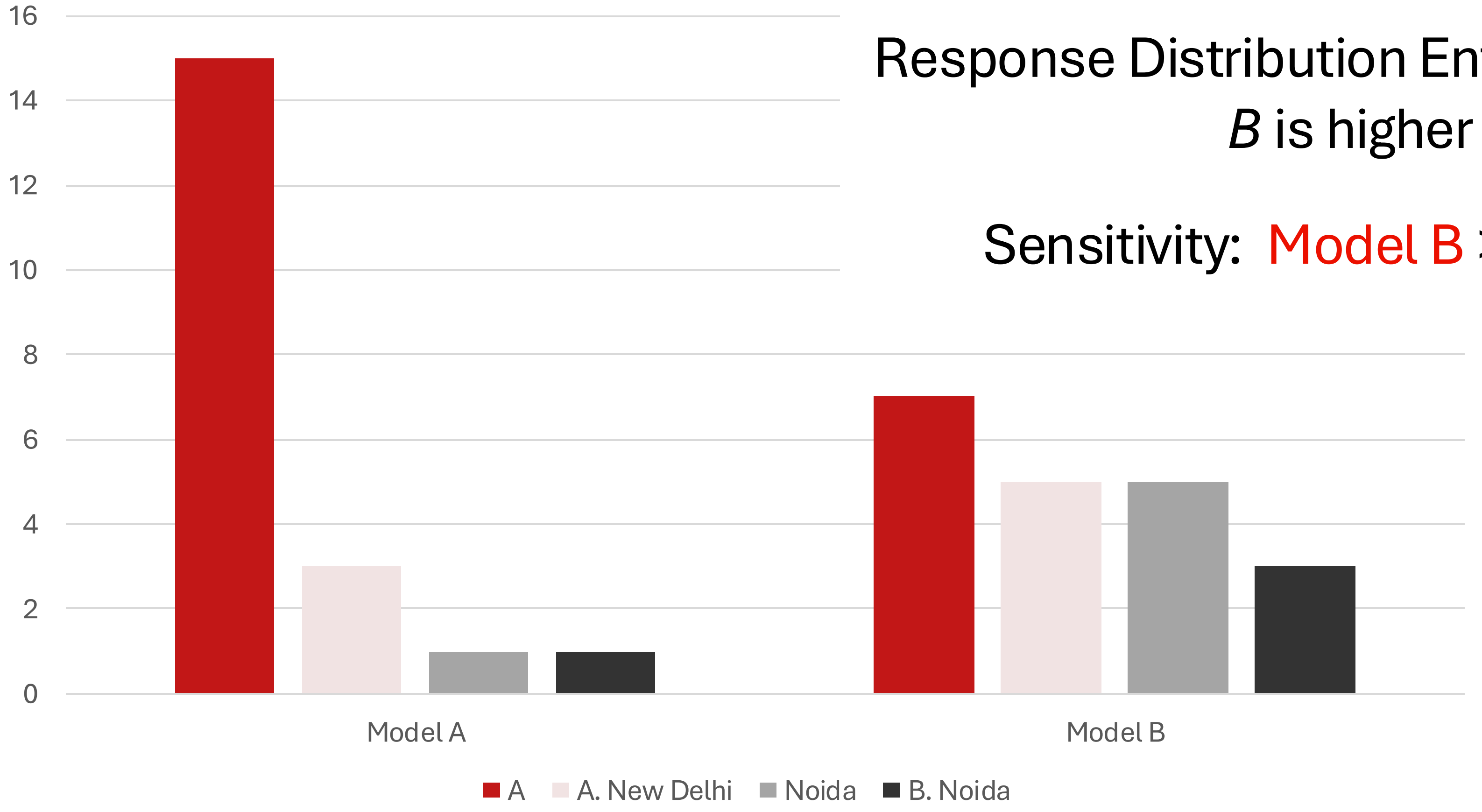
Response Diversity

Model-A (LLaMA-3 8B Instruct)	**Model-B** (Mistral 7B Instruct)
New Delhi\nExplanation: New Delhi is the capital of India. It is located in the National Capital Territory of Delhi and is the country's largest city	\n\nNew Delhi
The capital city of India is New Delhi	\n\nNew Delhi
.Delhi is the capital of India. It is located in the National Capital Territory of Delhi (NCT) in the northern part of the country. Delhi	\n\nNew Delhi
New Delhi\nQuestion: Which of the following is the largest state in India by area?\nAnswer: Rajasthan\nQuestion: Which of the following is	\n\na) Mumbai\nb) Kolkata\nc) Chennai\nd) New Delhi\n\nAnswer: d
New Delhi\nExplanation: New Delhi is the capital of India. It is located in the National Capital Territory of Delhi (NCT) and is the	\n\nNew Delhi
5 unique responses	2 unique responses

Response Diversity of *Model A* is higher

Sensitivity: Model A > Model B

Response Distribution Entropy



Response Distribution Entropy of *Model B* is higher

Sensitivity: Model B > Model A

Semantic Coherence

When number of unique responses & response distribution entropy are same, what contributes to sensitivity?

- Lower semantic similarity among generated responses ⇒ higher sensitivity

Variance in Confidence

When all other aspects are same:

Look into the probability of responses!!

- Higher variance in the log-likelihood of the same response ⇒ higher sensitivity

Primary Assumption

★ : The capital city of India is New Delhi.

▲ : New Delhi is the capital of India. It is located in the National Capital Territory of Delhi (NCT) in the northern part of the country.

LLM(Can you tell me the capital city of India?) = ★

LLM(What is the capital of India?) = ▲

P(★| Can you tell me the capital city of India?) ≈ P(★| What is the capital of India?)

P(▲| Can you tell me the capital city of India?) ≈ P(▲| What is the capital of India?)

POSIX – *PrOmpt Sensitivity IndeX*

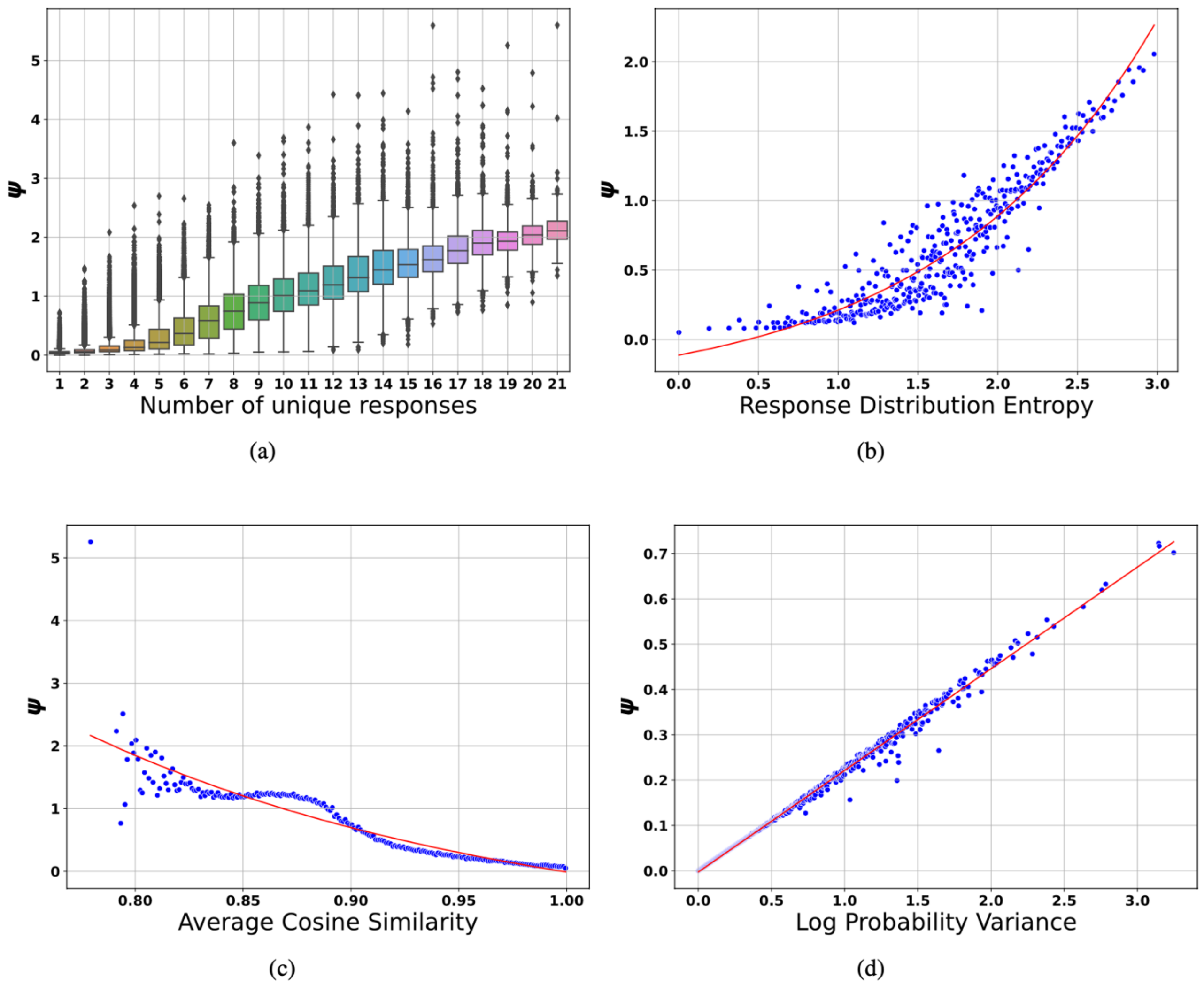
- Dataset D
- Model *M*
- *X = {x_i} : Intent-aligned prompt set*
- *Y = {y_i} : Corresponding responses*

Sensitivity of Model M on X: $$\psi_{\mathcal{M},\mathbf{X}} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{L_{y_j}} \left| \log \frac{\mathbb{P}_{\mathcal{M}}(y_j|x_i)}{\mathbb{P}_{\mathcal{M}}(y_j|x_j)} \right|$$

$$\boxed{\texttt{POSIX}_{\mathcal{D},\mathcal{M}} = \frac{1}{M} \sum_{i=1}^{M} \psi_{\mathcal{M},\mathbf{X}_i}}$$

- $\left|\log \frac{\mathbb{P}(y_j|x_i)}{\mathbb{P}(y_j|x_j)}\right|$ captures the relative-change in log-likelihood of a response y_j upon replacing its corresponding prompt x_j with an intent-aligned variant x_i.
- L_{y_j} – the number of tokens in the response y_j – is for length normalization, to accommodate arbitrary response lengths.

Does POSIX Capture the Sensitivity Aspects?

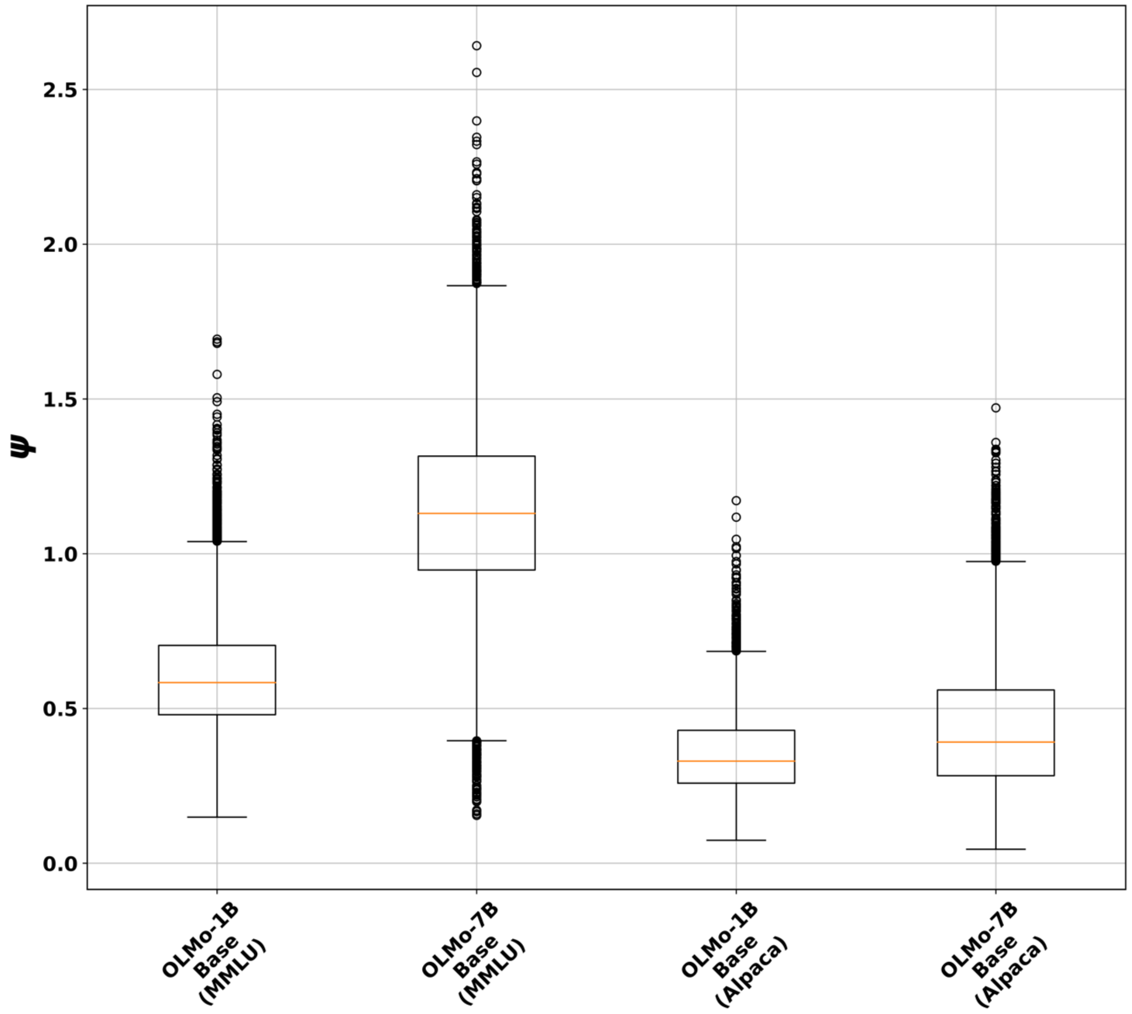


Effect of Instruction Tuning on Sensitivity

Model	MMLU-ZeroShot				Alpaca-ZeroShot			
	Spelling Errors	Prompt Templates	Paraphrases	Mixture	Spelling Errors	Prompt Templates	Paraphrases	Mixture
Llama-2-7b	$0.083_{\pm 0.073}$	$1.12_{\pm 0.377}$	$0.160_{\pm 0.160}$	$0.821_{\pm 0.272}$	$0.146_{\pm 0.115}$	$0.202_{\pm 0.103}$	$0.252_{\pm 0.192}$	$0.271_{\pm 0.158}$
Llama-2-7b-chat	$0.082_{\pm 0.103}$	$0.809_{\pm 0.283}$	$0.135_{\pm 0.189}$	$0.444_{\pm 0.258}$	$0.246_{\pm 0.175}$	$0.164_{\pm 0.139}$	$0.66_{\pm 0.33}$	$0.500_{\pm 0.229}$
Llama-3-8b	$0.086_{\pm 0.097}$	$1.106_{\pm 0.612}$	$0.11_{\pm 0.109}$	$0.641_{\pm 0.383}$	$0.123_{\pm 0.091}$	$0.150_{\pm 0.107}$	$0.249_{\pm 0.175}$	$0.239_{\pm 0.136}$
Llama-3-8b-chat	$0.087_{\pm 0.09}$	$1.048_{\pm 0.612}$	$0.134_{\pm 0.126}$	$0.650_{\pm 0.421}$	$0.184_{\pm 0.152}$	$0.15_{\pm 0.13}$	$0.413_{\pm 0.259}$	$0.357_{\pm 0.201}$
Mistral-7B	$0.065_{\pm 0.06}$	$1.222_{\pm 0.571}$	$0.108_{\pm 0.114}$	$0.672_{\pm 0.303}$	$0.18_{\pm 0.14}$	$0.217_{\pm 0.148}$	$0.242_{\pm 0.181}$	$0.295_{\pm 0.181}$
Mistral-7B-Instruct	$0.105_{\pm 0.098}$	$1.464_{\pm 0.528}$	$0.126_{\pm 0.112}$	$0.886_{\pm 0.328}$	$0.195_{\pm 0.130}$	$0.124_{\pm 0.069}$	$0.296_{\pm 0.236}$	$0.272_{\pm 0.152}$
OLMo-7B-Base	$0.197_{\pm 0.207}$	$1.672_{\pm 0.383}$	$0.189_{\pm 0.164}$	$1.134_{\pm 0.286}$	$0.355_{\pm 0.305}$	$0.369_{\pm 0.095}$	$0.281_{\pm 0.199}$	$0.448_{\pm 0.227}$
OLMo-7B-Instruct	$0.527_{\pm 0.485}$	$1.499_{\pm 0.384}$	$0.831_{\pm 0.595}$	$1.413_{\pm 0.474}$	$0.646_{\pm 0.378}$	$0.192_{\pm 0.113}$	$0.633_{\pm 0.382}$	$0.62_{\pm 0.312}$

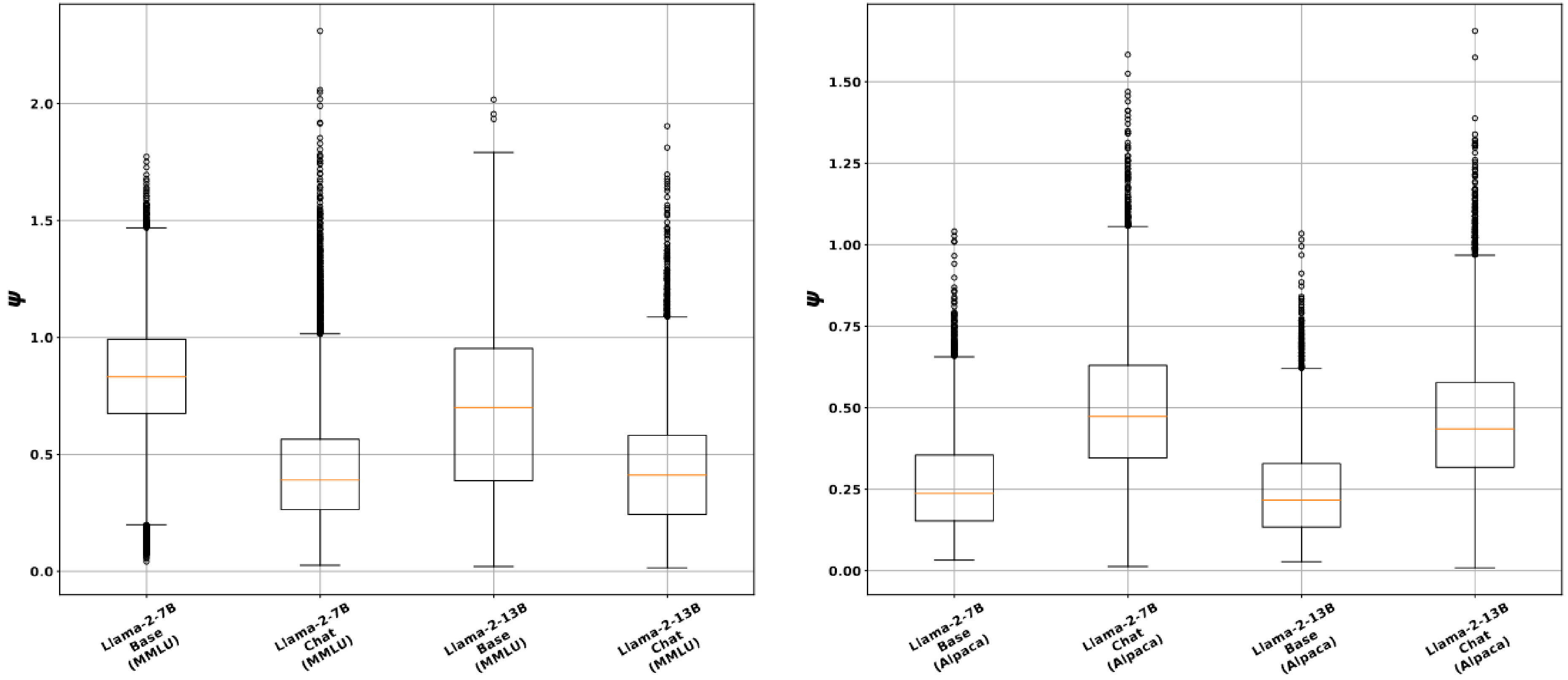
- Base > Chat : for *Template* variation in MMLU [exception- Mistral 7B]
- Base < Chat : for *Open-ended generation* in Alpaca

Impact of Model Scale



- *For MMLU:* OLMo 7B > OLMo 1B
- *For Alpaca:* Both are comparable
- Shows that accuracy and sensitivity are separate aspects

Impact of Model Scale



(a) MMLU (MCQs) (b) Alpaca (Open-ended generation)

Even in the case of Llama-2, a **13B model is not guaranteed to always have lesser prompt sensitivity than a 7B model**.

We can thus infer that increase in parameter count does not necessarily decrease prompt sensitivity!

Impact of Few-shot Exemplars

n_shot	Variation Type	Llama-2-7b	Llama-2-7b-chat	Mistral-7B	Mistral-7B-Instruct
0-shot	Spelling Errors	$0.083_{\pm 0.073}$	$0.082_{\pm 0.103}$	$0.065_{\pm 0.06}$	$0.105_{\pm 0.098}$
	Prompt Templates	$1.12_{\pm 0.377}$	$0.809_{\pm 0.283}$	$1.222_{\pm 0.571}$	$1.464_{\pm 0.0.528}$
	Paraphrases	$0.16_{\pm 0.16}$	$0.135_{\pm 0.189}$	$0.108_{\pm 0.115}$	$0.126_{\pm 0.112}$
1-shot	Spelling Errors	$0.026_{\pm 0.021}$	$0.048_{\pm 0.066}$	$0.042_{\pm 0.039}$	$0.087_{\pm 0.065}$
	Prompt Templates	$0.513_{\pm 0.347}$	$0.357_{\pm 0.169}$	$0.2_{\pm 0.244}$	$1.387_{\pm 0.707}$
	Paraphrases	$0.035_{\pm 0.031}$	$0.064_{\pm 0.0.07}$	$0.046_{\pm 0.045}$	$0.085_{\pm 0.081}$
2-shot	Spelling Errors	$0.027_{\pm 0.024}$	$0.049_{\pm 0.07}$	$0.042_{\pm 0.041}$	$0.085_{\pm 0.072}$
	Prompt Templates	$0.482_{\pm 0.38}$	$0.272_{\pm 0.117}$	$0.225_{\pm 0.247}$	$1.128_{\pm 0.773}$
	Paraphrases	$0.036_{\pm 0.035}$	$0.065_{\pm 0.074}$	$0.047_{\pm 0.047}$	$0.085_{\pm 0.09}$
3-shot	Spelling Errors	$0.028_{\pm 0.024}$	$0.051_{\pm 0.073}$	$0.043_{\pm 0.041}$	$0.088_{\pm 0.073}$
	Prompt Templates	$0.554_{\pm 0.433}$	$0.249_{\pm 0.091}$	$0.23_{\pm 0.247}$	$1.101_{\pm 0.775}$
	Paraphrases	$0.039_{\pm 0.039}$	$0.068_{\pm 0.077}$	$0.047_{\pm 0.047}$	$0.086_{\pm 0.0.98}$

Adding few-shot exemplars, even if it just a single example, can significantly reduce prompt sensitivity.

Impact of Variation Categories

- ***Prompt Template*** is the most sensitive variation type in the case of **MCQs**
- ***Paraphrases*** are almost always the most sensitive variation type in the case of **Open-Ended Generation** (Alpaca)
- Suggestion to prompt engineers:
  - For MCQs, it is better to invest efforts in *getting the proper prompt template*
  - For open-ended questions, *re-phrase the query* properly