In-context Learning

CSCI 601 471/671
NLP: Self-Supervised Models

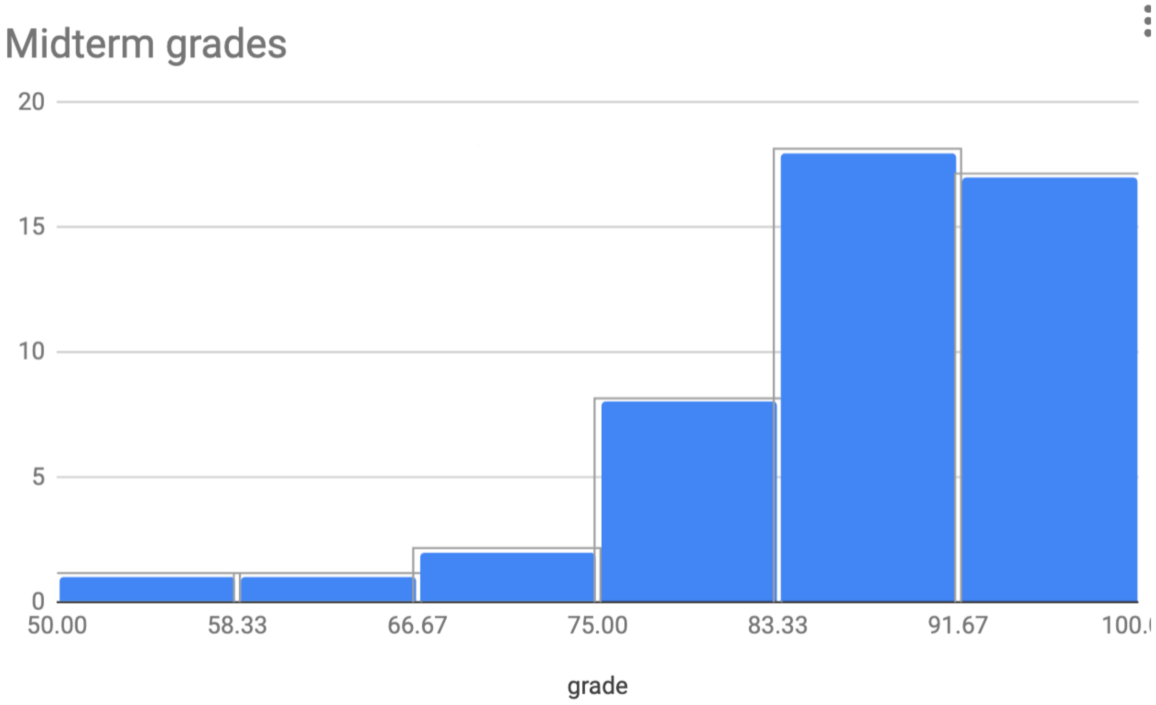
https://self-supervised.cs.jhu.edu/sp2023/

JOHNS HOPKINS UNIVERSITY

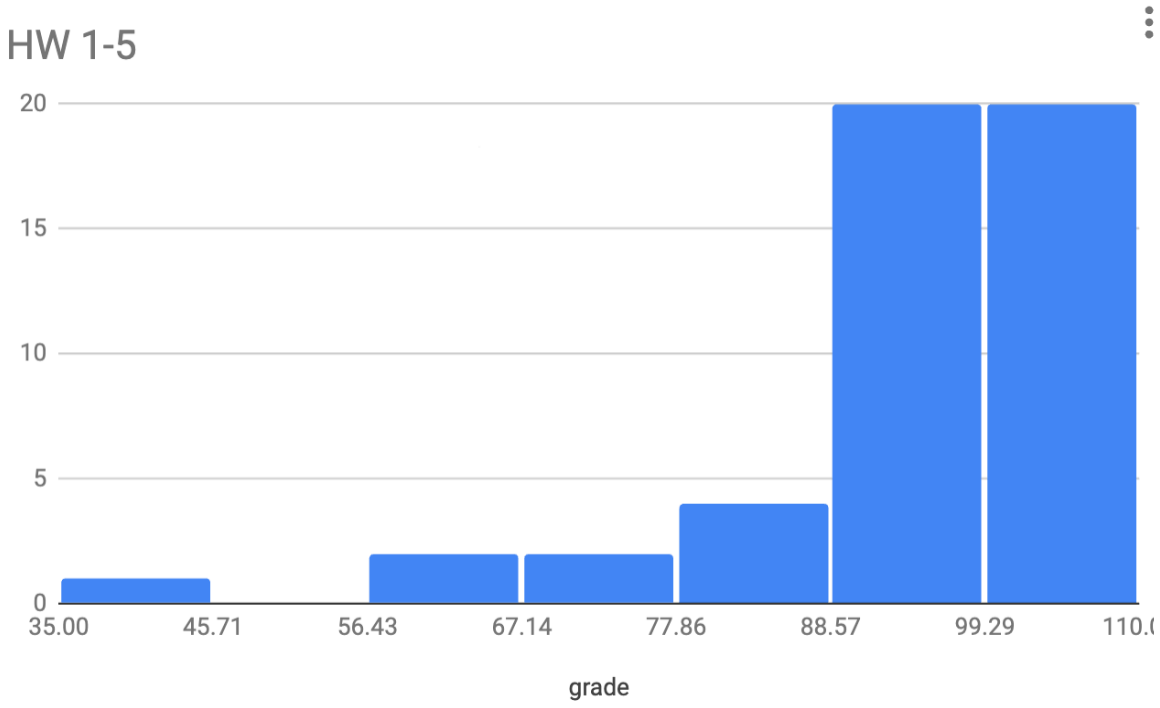
[Slide credit: Iz Beltagy, Arman Cohan, Robert Logan IV, Sewon Min, Sameer Singh, and many others]

Logistics

- HW7 is up! Due Thu March 30.
- Projects: Please continue to brainstorm!
  - Project proposal deadline: Thu March 30.
- Midterms grades?


Midterm grades
20
15
10
5
0
50.00
58.33
66.67
75.00
83.33
91.67
100.
grade

HW 1-5



News: GPT-4 was released!!

- "Transformer-style model pretrained to predict next token"
- We don't know the size ☹
- We don't know the amount of supervision ☹

> This report focuses on the capabilities, limitations, and safety properties of GPT-4. GPT-4 is a Transformer-style model [33] pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers. The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF) [34]. Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model si
> dataset construction, training method, or similar.

- From a company name "Open"AI — the irony

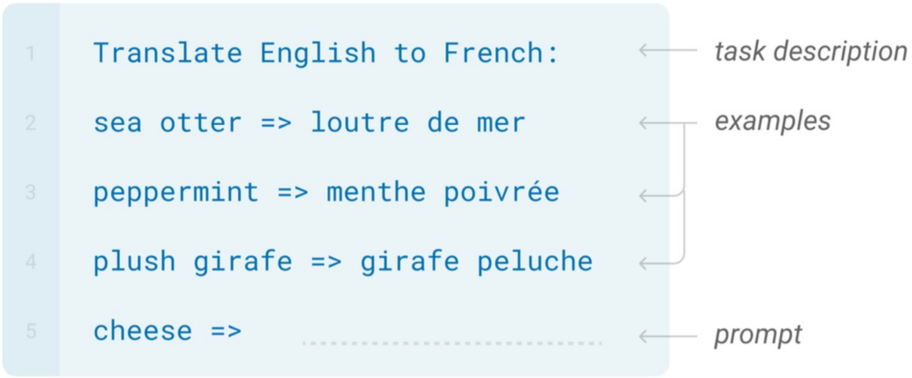
News: GPT-4 was released!!

> This report focuses on the capabilities, limitations, and safety properties of GPT-4. GPT-4 is a Transformer-style model [33] pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers. The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF) [34]. Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar.

- It is trained with human feedback (RLHF) — we will discuss it in a few weeks.
- It is trained on multi-modal signals — we will discuss it in a few weeks.

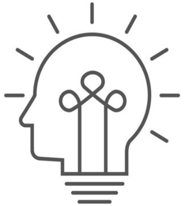
- More results in the technical report: https://cdn.openai.com/papers/gpt-4.pdf

In-Context Learning



- Learns to do a downstream task by conditioning on input-output examples!
- **No weight update** — our model is not **explicitly pre-trained** to learn from examples
  - The underlying models are quite general
- Today's focus:
  - How to use effectively in practice?
  - Fundamentally, why does it work?

How/Why does In-context Learning Work?



Any arbitrary task ⇨ Language Model

A few-shot learner

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

In-context Learning as Bayesian Inference

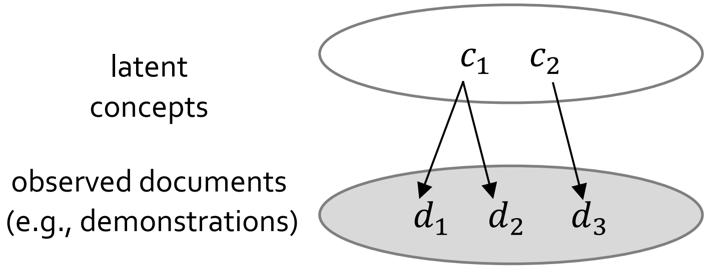
- (Xie et al., 2022) try to explain ICL as an implicit Bayesian inference.

Idea:

- (Pre-trained LM learn to represent "concepts", i.e. the ideas described by words.
- ICL enables LMs to "locate" the learned concepts.

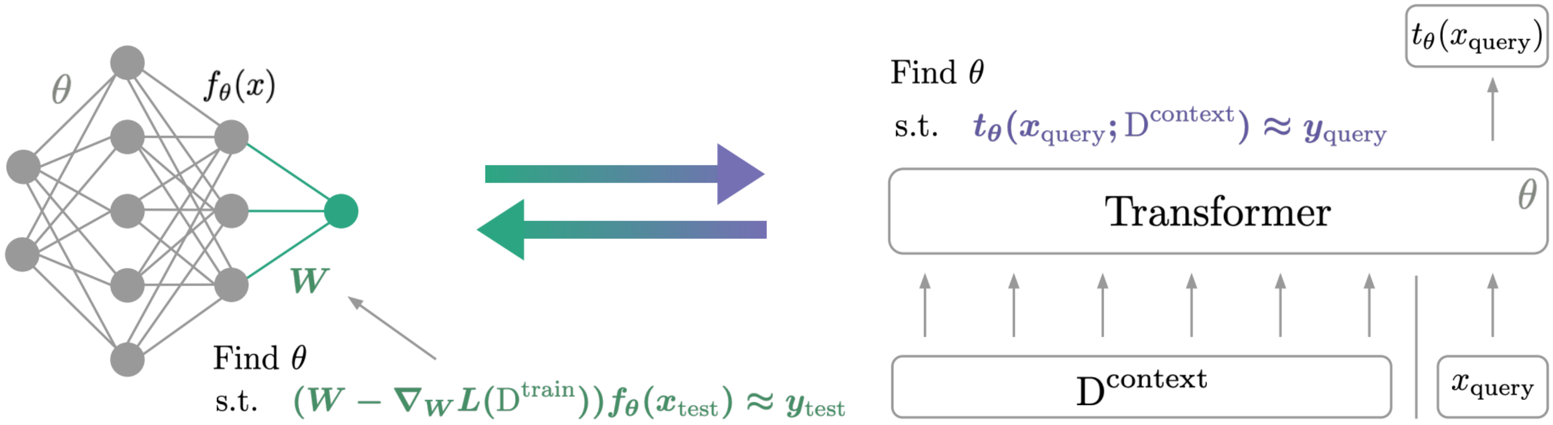
- Can formulate this intuition as a Bayesian inference
    - Prior over latent "concepts"
    - Likelihood describes connection between text and concepts
    - Given an incomplete doc, use Bayes formula to infer what concept is likely it is generated from and then complete the document.

- Does not explain everything.
    - GPT-3 can handle "unseen" concepts



In-context Learning as Gradience Descent

- ICL is implicitly equivalent to SGD on in-context demonstrations



[von Oswald et al. 2022; Akyurek et al. 2022; Dai et al. 2022, ...]

Summary & Open questions

- In-context learning has been a promising few-shot learning approach
  - No need for gradient updates ⟶ Much easier to use large models!

- Better calibration, better scoring of model outputs, and better formation of demonstrations lead to great improvements
  - How to make it less sensitive?
  - How to scale it (longer context, more training examples, wider range of tasks)?

- Still in progress ...
  - Understanding how/why it works,
  - Disentangling looking up task location vs learning a new task
  - Can we predict whether in-context learning would work on a given task or not?

Prompting for Multi-Step Reasoning

Some Problems Involve Reasoning

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

A: The answer is **5**

Arithmetic Reasoning (AR)
(+ −×÷…)

Q: Take the last letters of the words in "Elon Musk" and concatenate them

A: The answer is **nk.**

Symbolic Reasoning (SR)

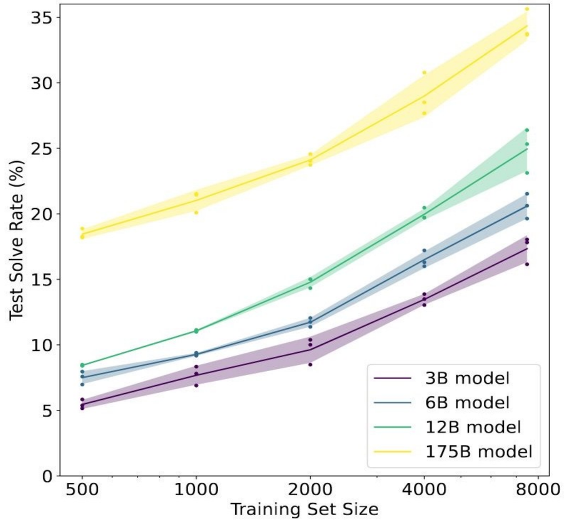
Q: What home entertainment equipment requires cable?
Answer Choices: (a) radio shack (b) substation (c) television (d) cabinet

A: The answer is **(c).**

Commonsense Reasoning (CR)

Reasoning Problems

(Cobbe et al. 2021)



- Fine-tune LMs on GSM8K (arithmetic reasoning)

- One may conjecture that, to achieve >80%, one needs 100x more training data for 175B model

- Another option is to increase model sizes, which is expensive.

- Other than these, how else can we improve the model performance on tasks that require multi-step reasoning?

Reasoning Problems via Multi-Step Prompting

- **Basic idea:** Rather than showing input-output pairs, prompting the model such that it shows its proof steps.

- **Note:** ideas around models that are capable of multi-step reasoning go way back.
  - Aristotle (deduction),
  - Hume (induction),
  - Peirce (abduction)
  - Lots of other works in pre-LM era
  - Namely, my Ph.D. thesis ☺ on multi-step reasoning in semantic representations of language

  [Reasoning-Driven Question-Answering for Natural Language Understanding]

- **Deduction**
  - All beans in that bag are white.
  - These beans are from that bag.
  - Therefore, these beans are white.
- **Induction**
  - These beans are from that bag.
  - These beans are white.
  - Therefore, all beans in that bag are white.
- **Abduction**
  - These beans are white.
  - All beans in that bag are white.
  - Therefore, these beans are from that bag.

Reasoning Problems via Multi-Step Prompting

(a) Few-shot

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: 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 answer is 8. **X**

Reasoning Problems via Multi-Step Prompting

(a) Few-shot

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: 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 answer is 8. X

(b) Few-shot-CoT （Wei et al., 2022）

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. ✓

Step-by-step demonstration

Step-by-step Answer

Reasoning Problems via Multi-Step Prompting

(a) Few-shot

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: 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 answer is 8. X

(b) Few-shot-CoT （Wei et al., 2022）

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. ✓

Step-by-step demonstration

Step-by-step Answer

(c) Zero-shot

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: The answer (arabic numerals) is

(Output) 8 X

Reasoning Problems via Multi-Step Prompting

(a) Few-shot

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: 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 answer is 8. ✗

(b) Few-shot-CoT (Wei et al., 2022)

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. ✓

Step-by-step demonstration

Step-by-step Answer

(c) Zero-shot

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: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (KoJima et al., 2022)

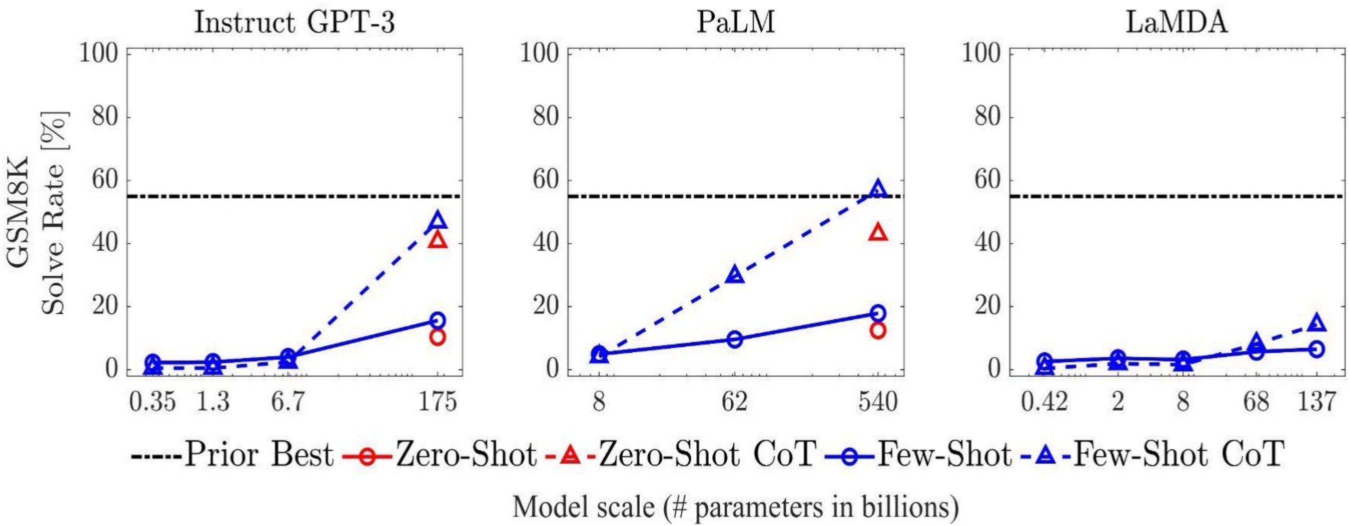
Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(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. ✓

Two-stage Prompting
Step-by-step Answer

Multi-Step Prompting: Empirical Results

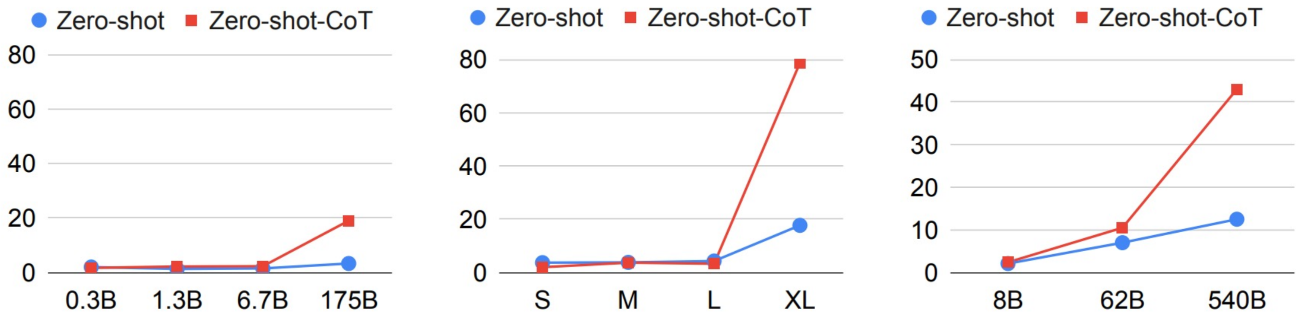
- **Setup:** show demonstrations that contain the decompositions
- The gains of multi-step prompting increases with scale.
- Prompting achieves better perf than [smaller] models that are fine-tuned on a lot more data.



["Chain of thought prompting elicits reasoning in large language models", Wei et al. 2022]

Multi-Step Prompting: Empirical Results

- **Setup:** show demonstrations that contain the decompositions
- The gains of multi-step prompting increases with scale.
- Prompting achieves better perf than [smaller] models that are fine-tuned on a lot more data.



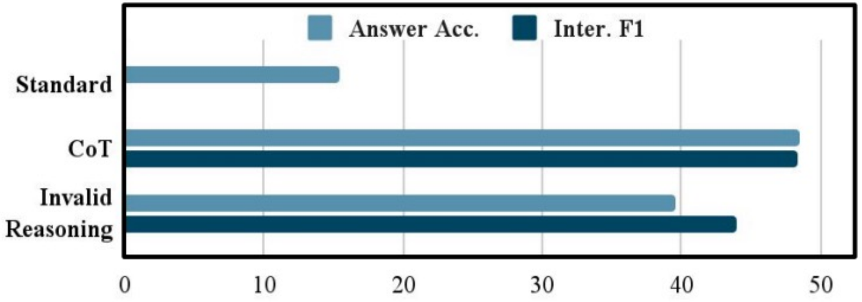
(a) MultiArith on Original GPT-3 (b) MultiArith on Instruct GPT-3 (c) GMS8K on PaLM

Figure 3: Model scale study with various types of models. S: text-ada-001, M: text-babbage-001, L: text-curie-001, XL: text-davinci-002. See Appendix A.3 and E for the detail.

["Large Language Models are Zero-Shot Reasoners", Kojima et al. 2022]

Multi-Step Prompting: Steps Don't Have to Be Correct!

- It is possible even with invalid demonstrations
- Prompting with invalid reasoning steps can achieve over 80-90% of the performance with correct reasoning steps.



	In-context Demonstration	Inference by LLM
Query	Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?	Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read?
Standard	The answer is 39.	The answer is 18. ✗
CoT	Originally, Leah had 32 chocolates and her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39 pieces left in total. The answer is 39.	Julie is reading a 120-page book. Yesterday, she read 12 pages and today, she read 24 pages. So she read a total of 12 + 24 = 36 pages. Now she has 120 - 36 = 84 pages left. Since she wants to read half of the remaining pages, she should read 84 / 2 = 42 pages. The answer is 42. ✓
Invalid Reasoning	Originally, Leah had 32 chocolates and her sister had 42. So her sister had 42 - 32 = 10 chocolates more than Leah has. After eating 35, since 10 + 35 = 45, they had 45 - 6 = 39 pieces left in total. The answer is 39.	Yesterday, Julie read 12 pages. Today, she read 12 * 2 = 24 pages. So she read a total of 12 + 24 = 36 pages. Now she needs to read 120 - 36 = 84 more pages. She wants to read half of the remaining pages tomorrow, so she needs to read 84 / 2 = 42 pages tomorrow. The answer is 42. ✓

[“Towards Understanding Chain-of-Thought Prompting”, Wang et al. 2022]

Multi-Step Prompting: Parting Comments

- Prompting LMs to explain their reasoning improves their performance.
- However, their steps aren't always correct.
    - A useful repository of annotation: https://github.com/OpenBioLink/ThoughtSource

- There is much to research on here:
    - When do LMs over-reason or under-reason?
    - How do adjust the granularity of step?
    - How to use use given references in the proofs?
    - How do use external "tools" (e.g., logic, calculator, Python) in forming proofs?

Parameter-Efficient Tuning of LMs

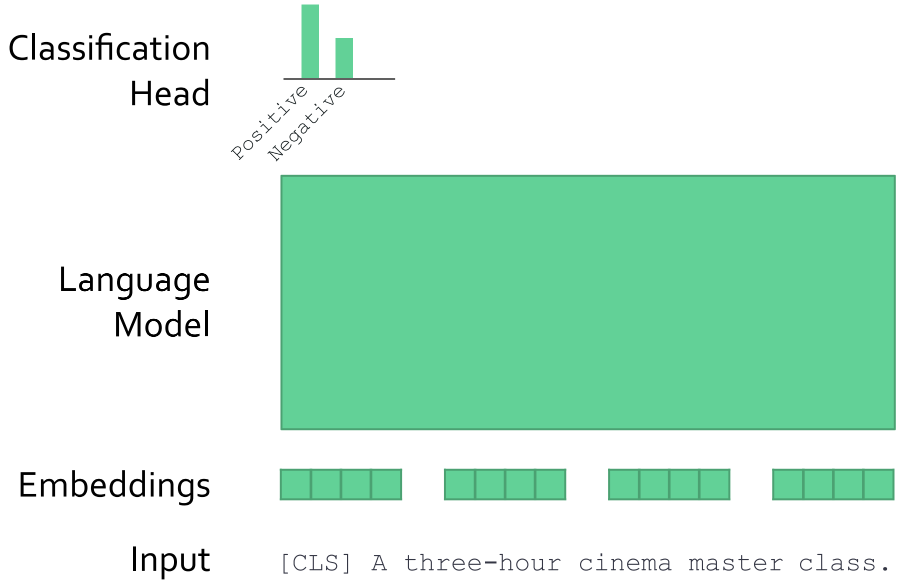
CSCI 601 471/671
NLP: Self-Supervised Models

https://self-supervised.cs.jhu.edu/sp2023/

JOHNS HOPKINS UNIVERSITY

[Slide credit: Iz Beltagy, Arman Cohan, Robert Logan IV, Sewon Min, Sameer Singh, Danqi Chen and many others]

Fine-tuning Pre-trained Models

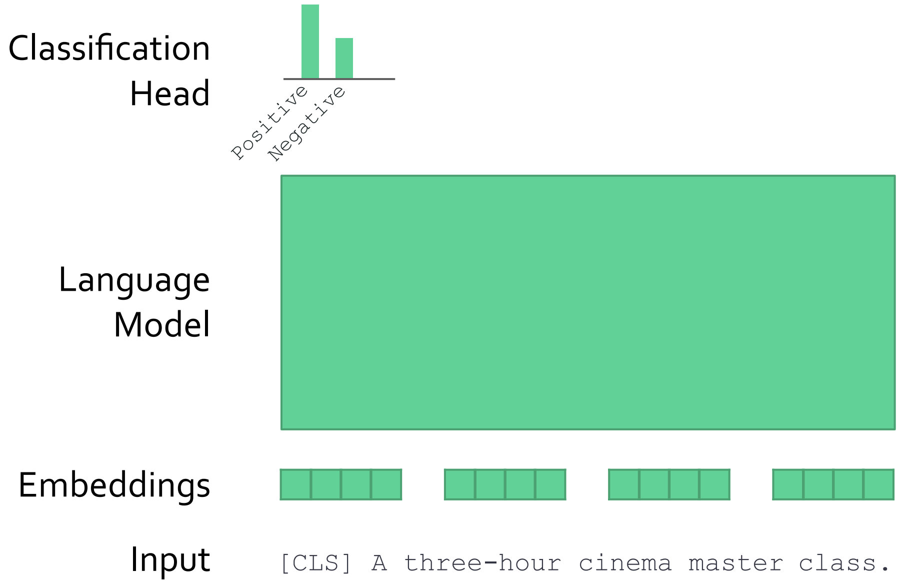


A general recipe:

- Pre-train a language model
- Fine-tune a classification head on top of the LMs representations

[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Fine-tuning Pre-trained Models



Default finetuning recommendations are unstable in few-shot settings.

Stability can be improved by:

- Using smaller learning rates
- Training for more iterations
- ...

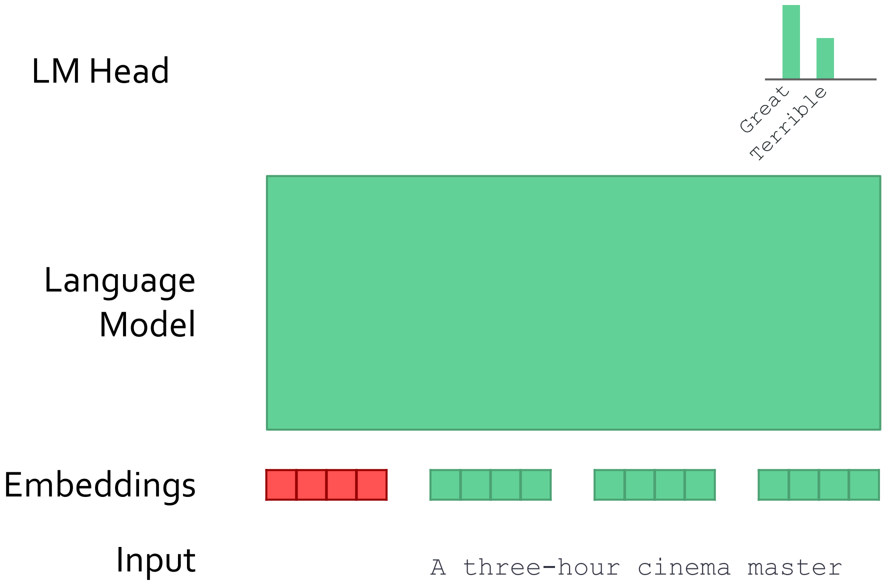
However finetuning still underperforms other methods.

["Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping" Dodge et al., 2020]
["On the Stability of Fine-tuning BERT:Misconceptions, Explanations, and Strong Baselines" Mosbach et al., 2020.]
["Revisiting Few-sample BERT Fine-tuning" Zhang et al., 2020]
[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Prompt Tuning

- Learn embeddings for placeholder tokens in the pattern.

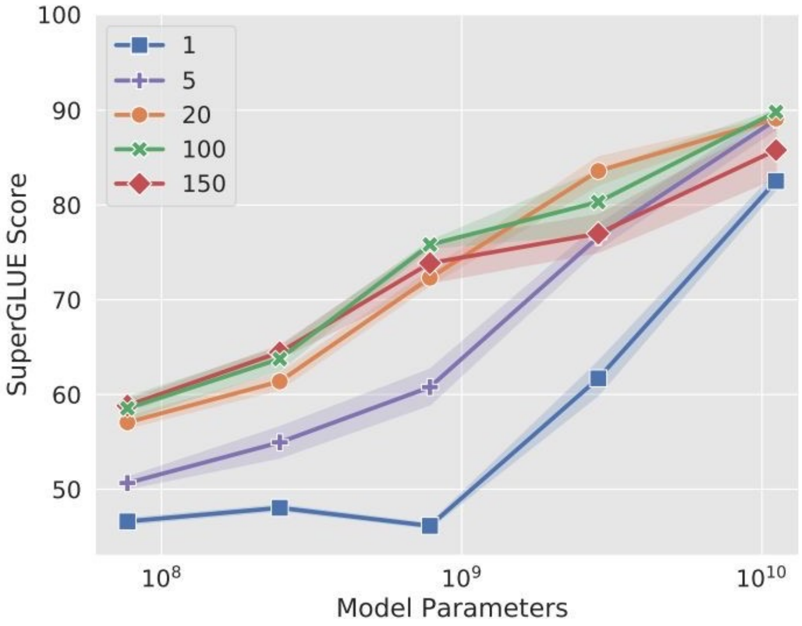
- Variants:
  - WARP [Hambardzumyan et al., 2021]
  - OptiPrompt [Zhong et al., 2021]
  - Prompt Tuning [Lester et al., 2021]
  - P-Tuning* [Li et al., 2021]



[ACL 2022 Tutorial Beltagy, Cohan, Logan IV, Min and Singh]

Prompt Tuning: Effect of Prompt Length

- The shorter the prompt, the fewer new parameters must be tuned
- Increasing prompt length is critical to achieving good performance
- The largest model still gives strong results with a single-token prompt
- Increasing beyond 20 tokens only yields marginal gains



[The Power of Scale for Parameter-Efficient Prompt Tuning. Lester et al. 2021]

BitFit

- BitFit adds bias terms in self-attention and MLP layers and tunes those.

$$\mathbf{Q}^{m,\ell}(\mathbf{x}) = \mathbf{W}_q^{m,\ell}\mathbf{x} + \mathbf{b}_q^{m,\ell}$$
$$\mathbf{K}^{m,\ell}(\mathbf{x}) = \mathbf{W}_k^{m,\ell}\mathbf{x} + \mathbf{b}_k^{m,\ell}$$
$$\mathbf{V}^{m,\ell}(\mathbf{x}) = \mathbf{W}_v^{m,\ell}\mathbf{x} + \mathbf{b}_v^{m,\ell}$$

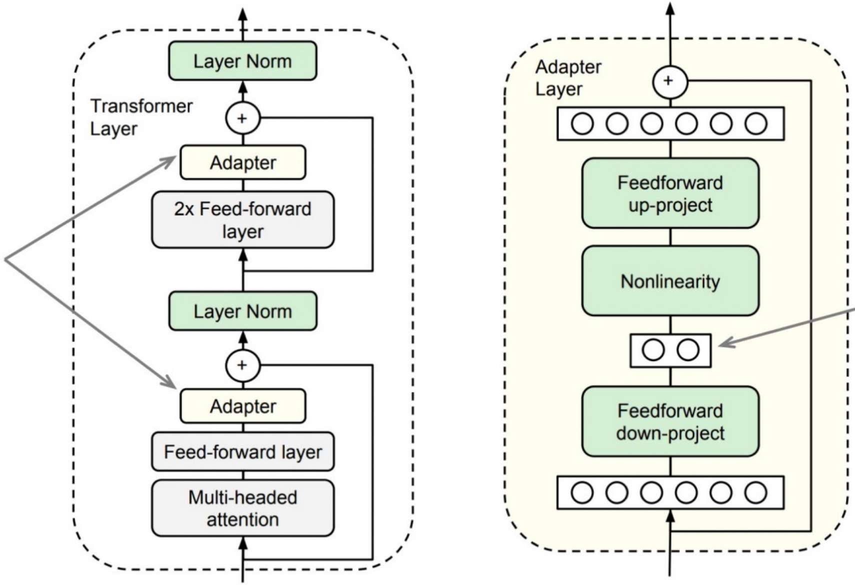
$$\mathbf{h}_2^\ell = \text{Dropout}\left(\mathbf{W}_{m_1}^\ell \cdot \mathbf{h}_1^\ell + \mathbf{b}_{m_1}^\ell\right) \quad (1)$$
$$\mathbf{h}_3^\ell = \mathbf{g}_{LN_1}^\ell \odot \frac{(\mathbf{h}_2^\ell + \mathbf{x}) - \mu}{\sigma} + \mathbf{b}_{LN_1}^\ell \quad (2)$$
$$\mathbf{h}_4^\ell = \text{GELU}\left(\mathbf{W}_{m_2}^\ell \cdot \mathbf{h}_3^\ell + \mathbf{b}_{m_2}^\ell\right) \quad (3)$$
$$\mathbf{h}_5^\ell = \text{Dropout}\left(\mathbf{W}_{m_3}^\ell \cdot \mathbf{h}_4^\ell + \mathbf{b}_{m_3}^\ell\right) \quad (4)$$
$$\text{out}^\ell = \mathbf{g}_{LN_2}^\ell \odot \frac{(\mathbf{h}_5^\ell + \mathbf{h}_3^\ell) - \mu}{\sigma} + \mathbf{b}_{LN_2}^\ell \quad (5)$$

["BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models" Ben Zaken et al., 2021.]

Adapters

- **Core idea:** train small sub-networks and only tune those.
- No need to store a full model for each task, only the adapter params.

Only these are trained, everything else is fixed and is the same for all tasks

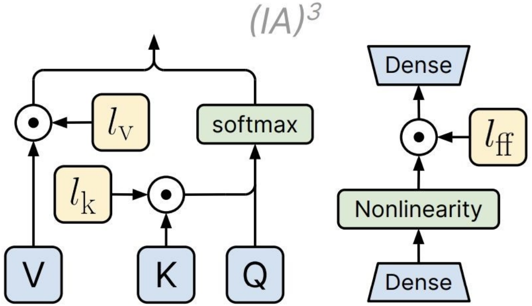


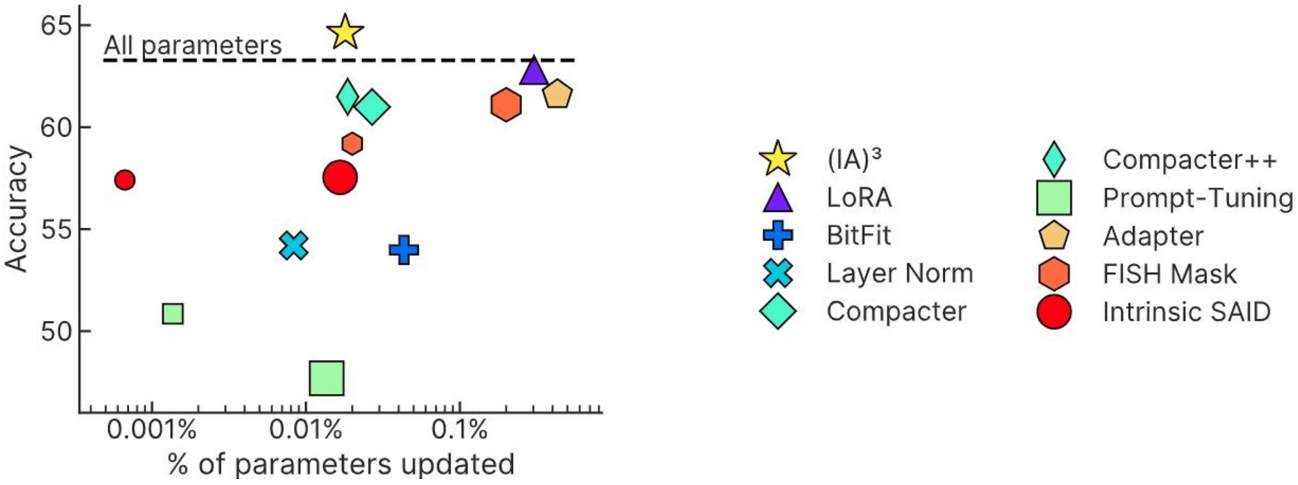
Small hidden size, i.e. an adaptor has only a few parameters (which is good!)

["Parameter-Efficient Transfer Learning for NLP" Houlsby et al., 2019.]

(IA)^3: Infused Adapter by Inhibiting and Amplifying Inner Activations

- Element-wise rescaling of model activations with a learned vector:
  - keys and values in self-attention
  - feed-forward networks



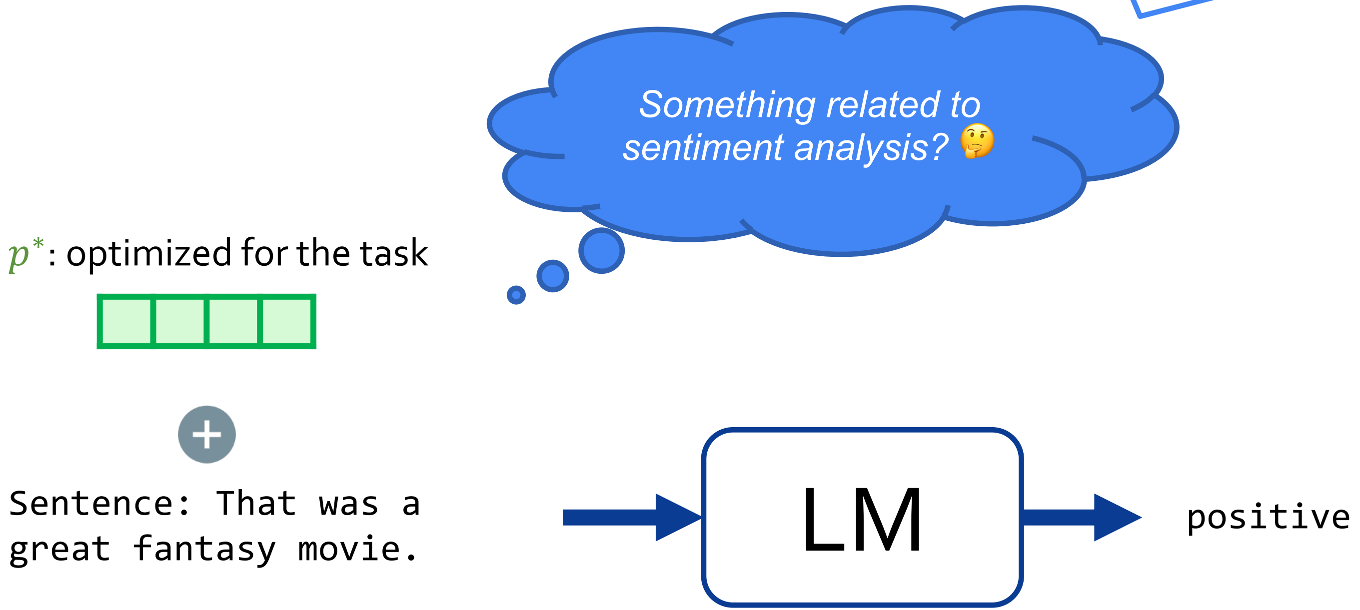


["Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning" Liu et al., 2022.]

Prompt Tuning: Interpretability

- Are continuous prompts interpretable?

Opposite goal: how unfaithful can their interpretation be of what they do?



["Prompt Waywardness: The Curious Case of Discretized Interpretation of Continuous Prompts" Khashabi et al., 2022.]

nearest-neighbor
mapping of continuous prompt
onto the word embeddings

$\tilde{p}$: optimized for the task + project to a given text

p^*: optimized for the task

definition of another task:

```
Write down the conclusion you can
reach by combining the given
Fact 1 and Fact 2.
```

random sentence from web:

```
int clamp(int val, int min_val) {
    return std::max(min_val, val);
}
```

+

```
Sentence: That was a
great fantasy movie.
```

LM

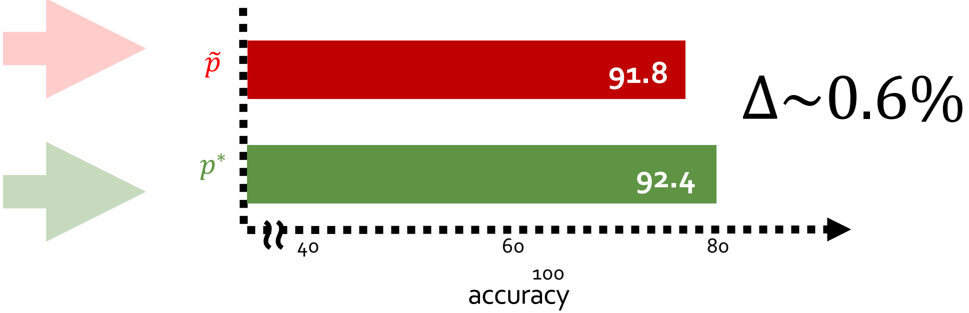
```
positive
```

["Prompt Waywardness: The Curious Case of Discretized Interpretation of Continuous Prompts" Khashabi et al., 2022.]

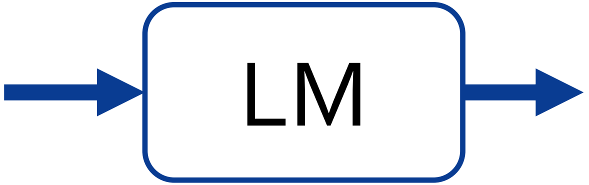
continuous prompts that
project to any given text
with tiny drop in task accuracy!

$\tilde{p}$: optimized for the task + project to a given text

p^*: optimized for the task

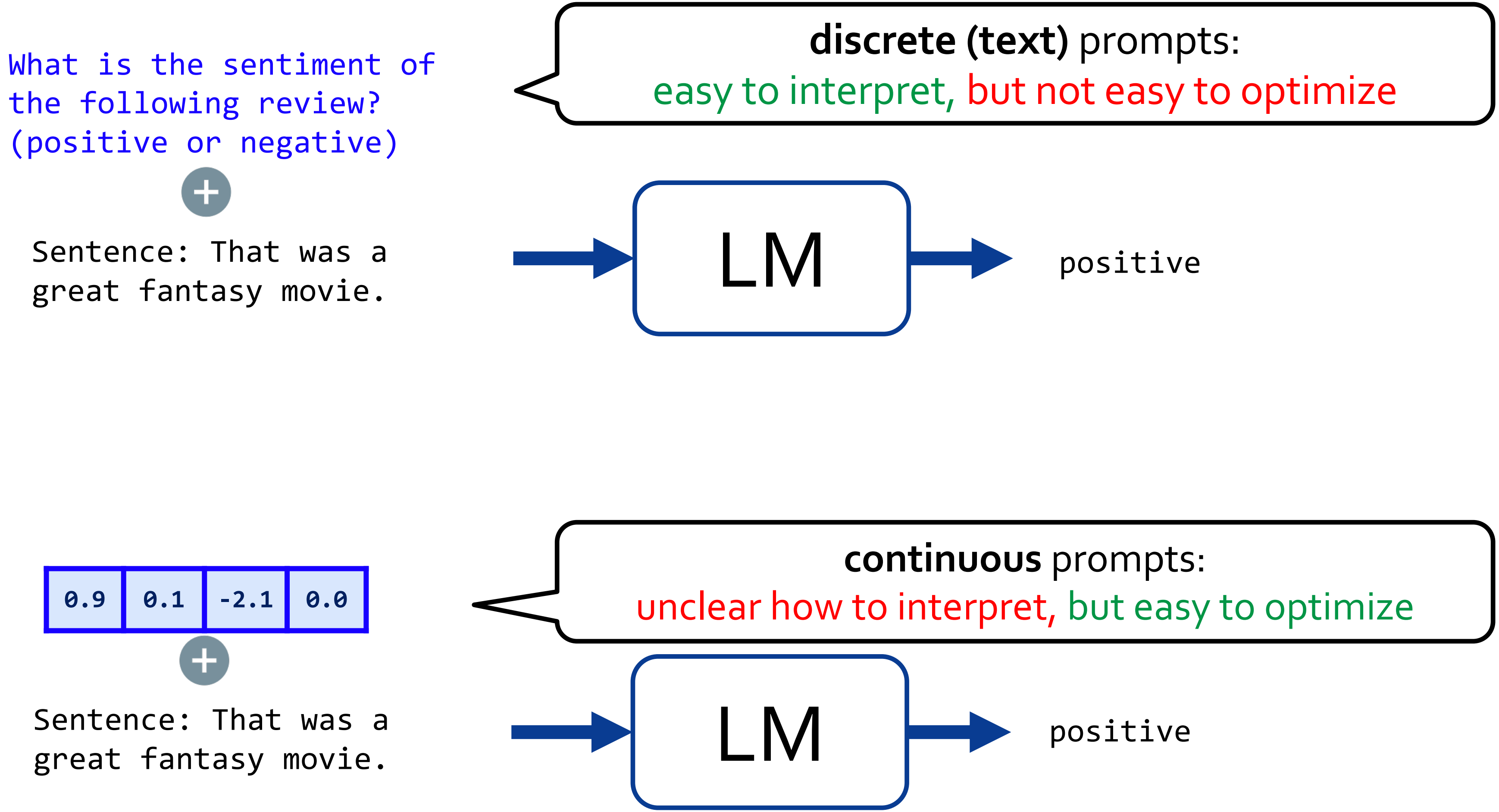


`Sentence: That was a great fantasy movie.`



`positive`

["Prompt Waywardness: The Curious Case of Discretized Interpretation of Continuous Prompts" Khashabi et al., 2022.]



["Prompt Waywardness: The Curious Case of Discretized Interpretation of Continuous Prompts" Khashabi et al., 2022.]

Open questions & future work

- Parameter efficient optimization — optimize fewer parameters than the whole model.
  - Space efficiency — fewer parameters to store
  - Computation efficiency? A bit unclear
- Their interpretability is not quite clear.
- **Open research question:** How to bridge the gap between continuous prompts vs discrete prompts?