Prompting and In-Context Learning with Large Language Models

Sewon Min
University of Washington
https://shmsw25.github.io | sewon@cs.washington.edu

March 6, 2023, CSE 447

In this lecture...

- Prompting & In-context learning
- Terminologies
- Improving prompting/in-context learning
- Understanding prompting/in-context learning
- Takeaways

Prompting & In-Context Learning

Prompting:
Using a large language model
to perform a new task
without gradient updates

Prompting:
Using a large language model
to perform a new task
without gradient updates

Prompting:

Using a large language model to perform a new task without gradient updates

Task

A three-hour cinema master class. → positive / negative

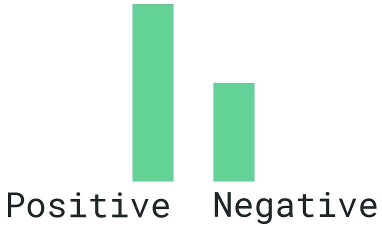
Supervised learning

Labeled Training Data

“An effortlessly accomplished and richly resonant work”: Positive

“A mostly tired retread of several other mob tales.”: Negative

....



Some neural model (RNN, LSTM, Transformer)

A three-hour cinema master class.

Language Models

Internet data

I am remarkably stingy with my 10/10 ratings. I'll be the first person to acknowledge this. Of the roughly 2600 titles I've rated on here, only 34 have a 10. Parasite is one of them. If this isn't a masterpiece, then I don't know what is. I'm going to keep it vague on the plot-front, because I didn't know anything about it going in, and was really excited to see it progress and unfold in satisfying, unexpected ways. (...)

ratings

Language Model

I am remarkably stingy with my 10/10 ______

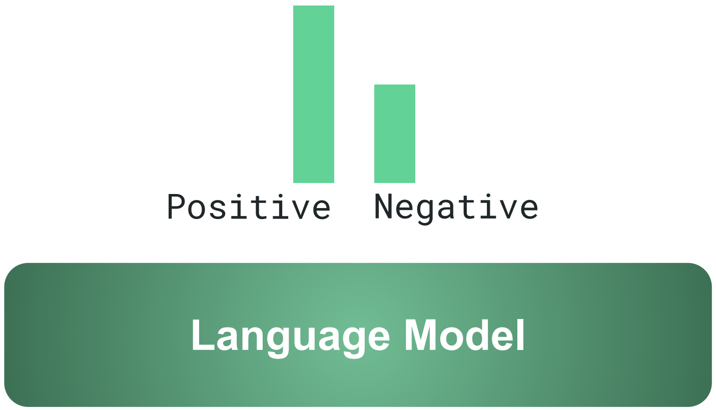
Fine-tuning

Labeled Training Data

“An effortlessly accomplished and richly resonant work”: Positive

“A mostly tired retread of several other mob tales.”: Negative

....



A three-hour cinema master class.

Perform the task without finetuning,
without large training data for the task of interest?

LM Prompting

great

terrible

(Frozen) **Language Model**

A three-hour cinema master class. It was ______

P1 = P(It was **great**! | A three-hour cinema master class.)

P2 = P(It was **terrible**! | A three-hour cinema master class.)

P1>P2 "**positive**"

P1<P2 "**negative**"

Zero-shot

Brown et al. 2020. "Language Models are Few-Shot Learners"

In-context Learning (GPT3; Brown et al., 2020)

Movie review dataset

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

In-context Learning (GPT3; Brown et al., 2020)

Movie review dataset

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

An effortlessly accomplished and richly resonant work. It was great!

In-context Learning (GPT3; Brown et al., 2020)

Movie review dataset

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

An effortlessly accomplished and richly resonant work. It was great!

A mostly tired retread of several other mob tales. It was terrible!

In-context Learning (GPT3; Brown et al., 2020)

Movie review dataset

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

An effortlessly accomplished and richly resonant work. It was great!

+

A mostly tired retread of several other mob tales. It was terrible!

+

Test input → A three-hour cinema master class. It was ________

In-context Learning (GPT3; Brown et al., 2020)

An effortlessly accomplished and richly resonant work. It was great!

A mostly tired retread of several other mob tales. It was terrible!

A three-hour cinema master class. It was ________

Language Model ?

great

terrible

P1 = P(It was **great**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

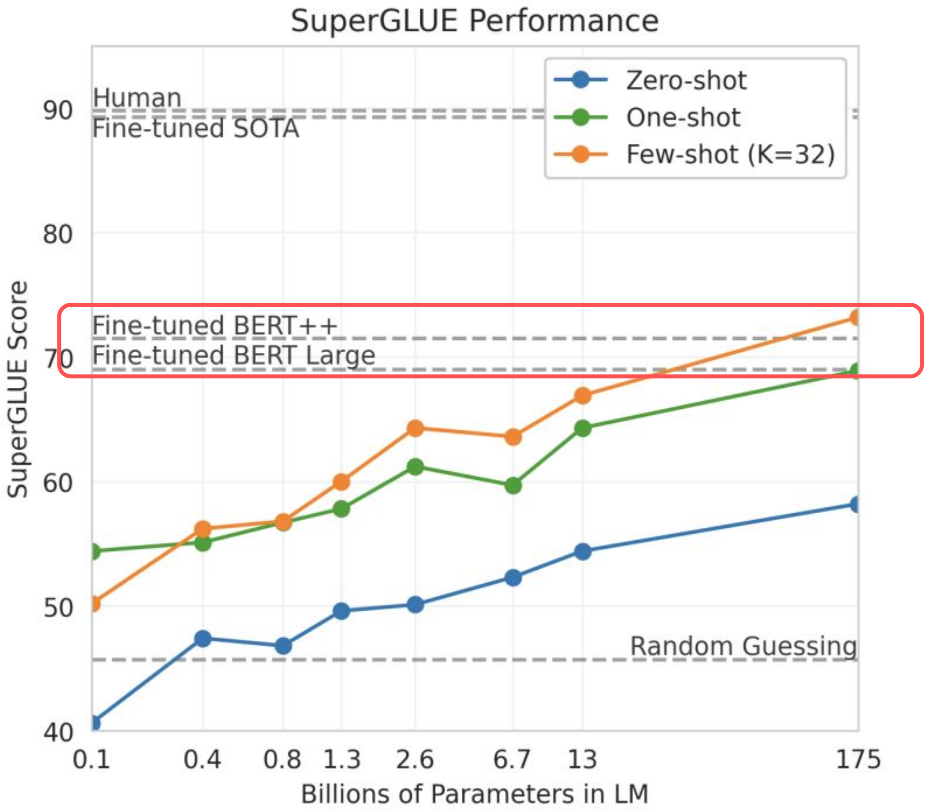
P2 = P(It was **terrible**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

P1>P2 "**positive**"

P1<P2 "**negative**"

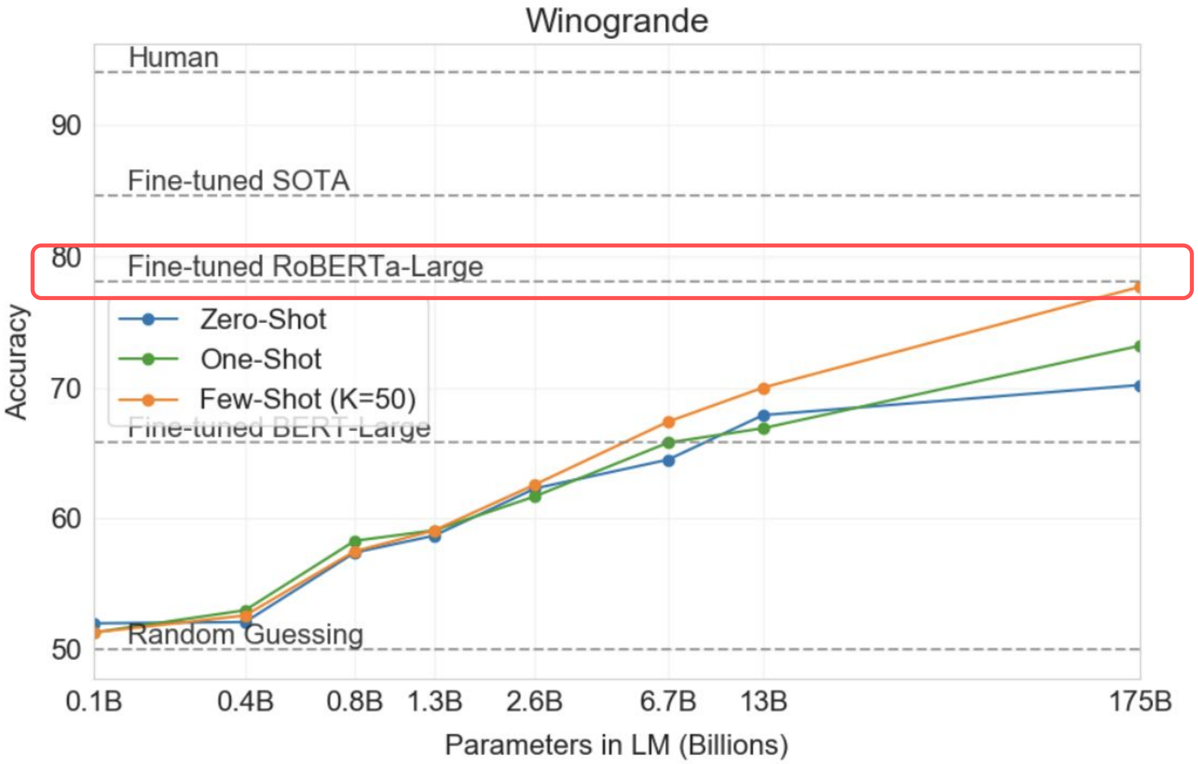
Few-shot / *k*-shot

In-context learning results



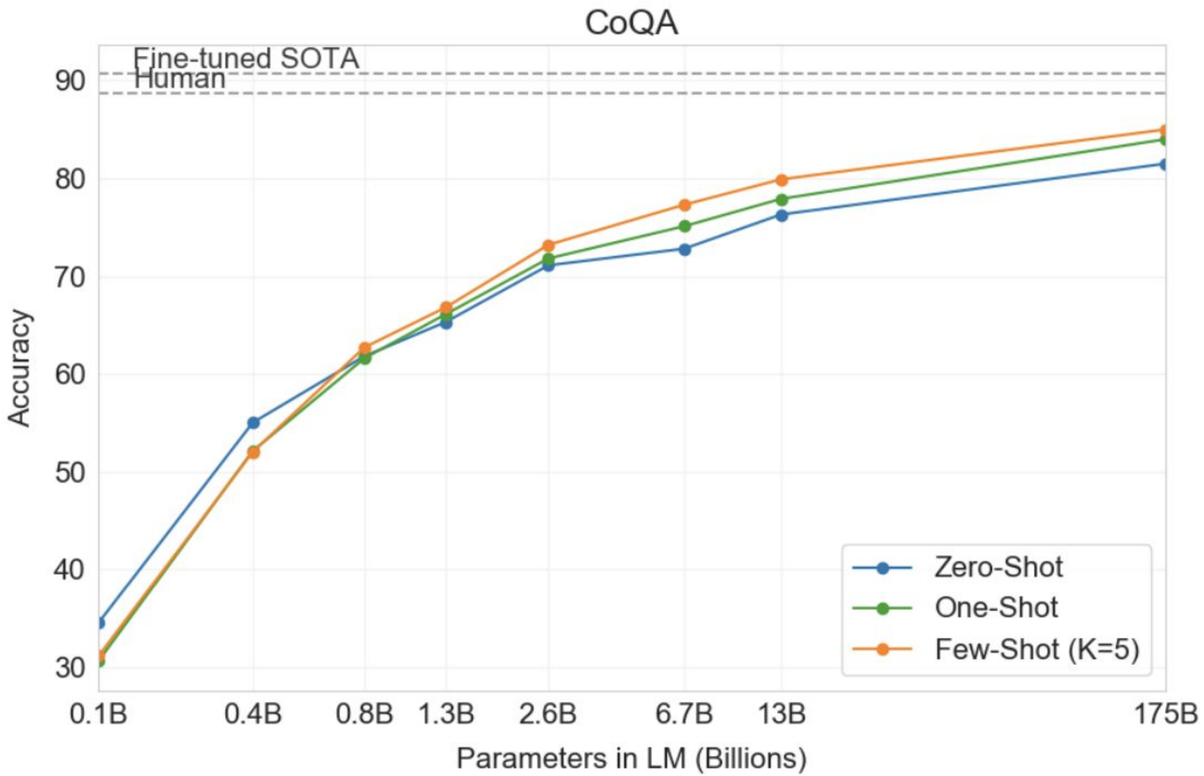
Brown et al. 2020. “Language Models are Few-Shot Learners”

In-context learning results



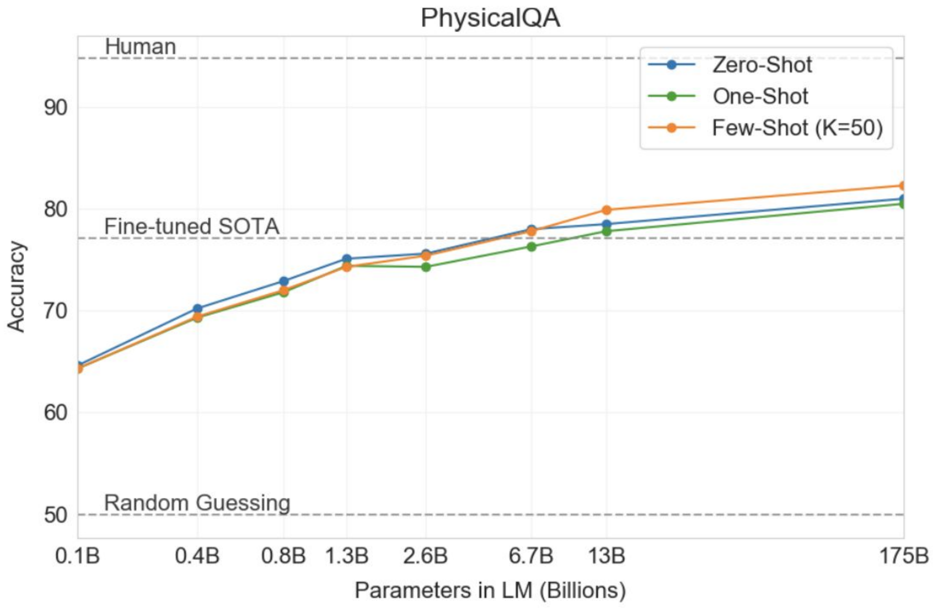
Brown et al. 2020. “Language Models are Few-Shot Learners”

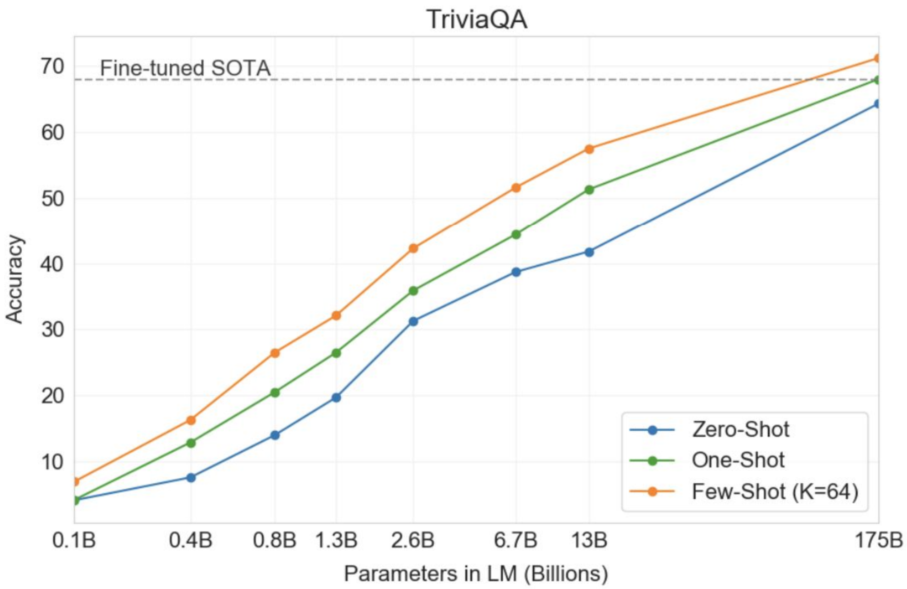
In-context learning results



Brown et al. 2020. “Language Models are Few-Shot Learners”

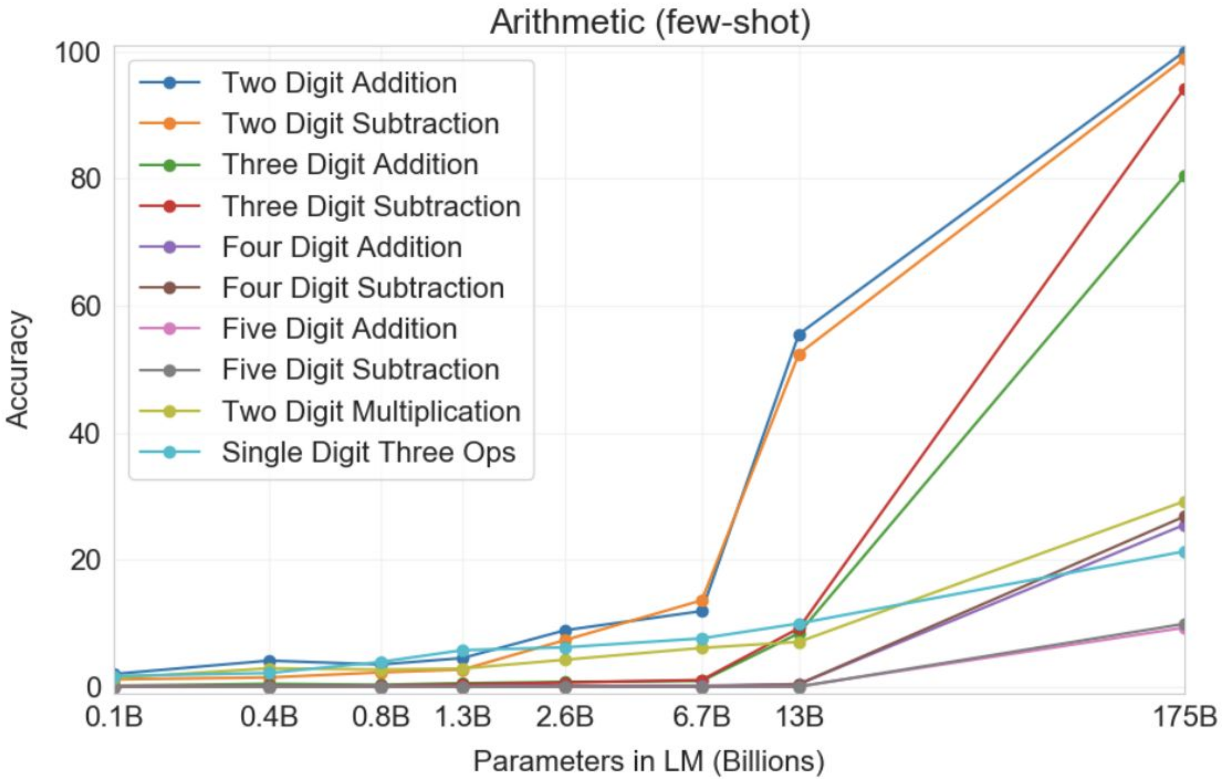
In-context learning results





[Brown et al. 2020](). "Language Models are Few-Shot Learners"

In-context learning results

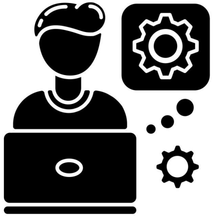


Brown et al. 2020. “Language Models are Few-Shot Learners”

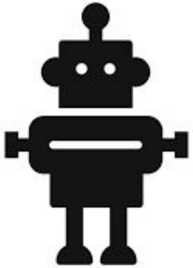
Why is it amazing?

No need to collect large labeled data

No need to do gradient updates



Scientifically interesting
(Closer to *fundamental intelligence?*)



Terminologies

Input to the LM

An effortlessly accomplished and richly resonant work.	It was great!
A mostly tired retread of several other mob tales.	It was terrible!
A three-hour cinema master class.	It was ______!

***Prompt*:** A conditioning text coming before the test input

Demonstrations: A special instance of prompt which is a concatenation of the k-shot training data (in in-context learning, prompt==demonstrations)

(Prompt may be different from demonstrations outside of in-context learning, e.g., a description about the task).

Terminologies

Input to the LM

An effortlessly accomplished and richly resonant work. | It was great!

A mostly tired retread of several other mob tales. | It was terrible!

A three-hour cinema master class. | It was ______!

Prompt: A conditioning text coming before the test input

Demonstrations: A special instance of prompt which is a concatenation of the k-shot training data (in in-context learning, prompt==demonstrations)

Pattern: A function that maps an input to the text (a.k.a. template)

Verbalizer: A function that maps a label to the text (a.k.a. label words)

Examples of patterns/verbalizers

An effortlessly accomplished and richly resonant work.	It was great!
A mostly tired retread of several other mob tales.	It was terrible!
A three-hour cinema master class.	It was great!

Pattern: f(<x>) = <x>
Verbalizer: v(“positive”) = “It was great!”, f(“negative”) = “It was terrible!”

Review: An effortlessly accomplished and richly resonant work.	Sentiment: positive
Review: A mostly tired retread of several other mob tales.	Sentiment: negative
Review: A three-hour cinema master class.	Sentiment: positive

Pattern: f(<x>) = “Review: <x>”
Verbalizer: v(<x>) = “Sentiment: <x>”

Notes on patterns/verbalizers

- There are many different possible patterns/verbalizers even for the same task.
- In practice, it is better to use patterns/verbalizers that makes the sequence closer to language modeling, i.e. closer to the text that the model might have seen during pretraining.
- It turns out there is huge variance in performance based on the choice of patterns/verbalizers (more in the next slide).
- You should not choose patterns/verbalizers based on the test data.

Review

Test data: (x, y) **Train data:** $(x_1, y_1, \cdots, x_k, y_k)$ **Pattern:** f **Verbalizer:** v

Zero-shot prompting: $\text{argmax}_{y \in \mathcal{Y}} P_{\text{LM}}(v(y) | f(x))$

In-context learning: $\text{argmax}_{y \in \mathcal{Y}} P_{\text{LM}}(v(y) | f(x_1), v(y_1), \cdots, f(x_k), v(y_k), f(x))$

For simplicity, from now on…

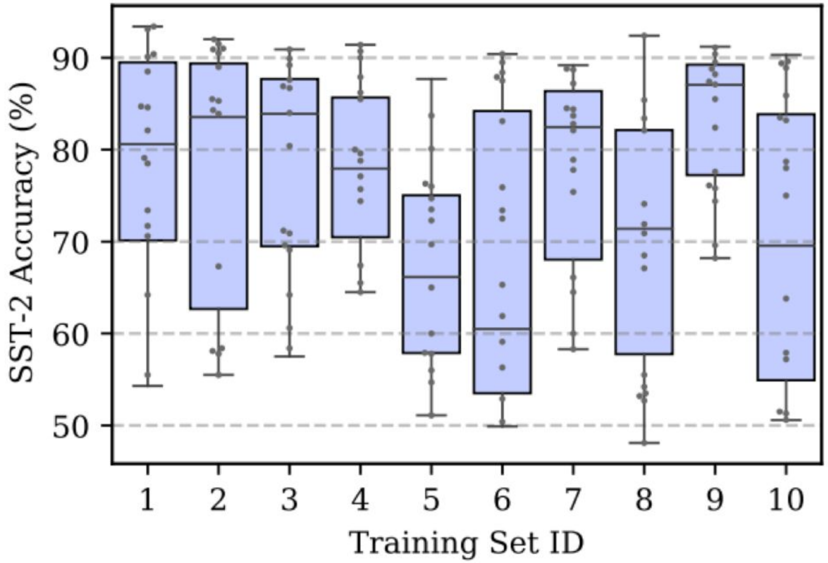
Zero-shot prompting: $\text{argmax}_{y \in \mathcal{Y}} P_{\text{LM}}(y | x)$

In-context learning: $\text{argmax}_{y \in \mathcal{Y}} P_{\text{LM}}(y | x_1, y_1, \cdots, x_k, y_k, x)$

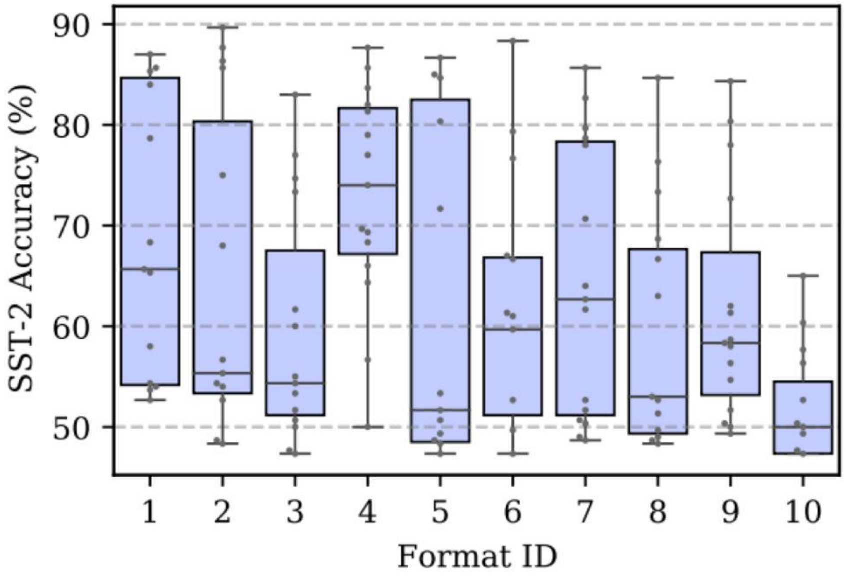
Limitations &
How to improve them?

Variance

Across different training sets and permutations



Across different training sets and patterns/verbalizers



Zhao et al. 2021. “Calibrate Before Use: Improving Few-Shot Performance of Language Models”

Variance

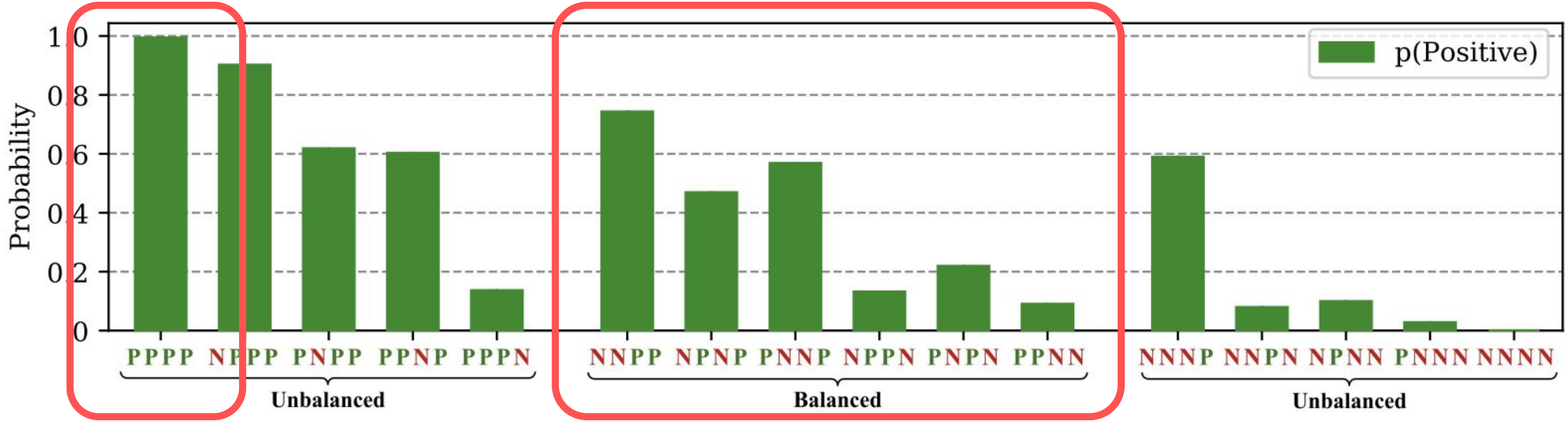


Figure 4. **Majority label and recency biases** cause GPT-3 to become biased towards certain answers and help to explain the high variance across different examples and orderings. Above, we use 4-shot SST-2 with prompts that have different class balances and permutations, e.g., [P P N N] indicates two positive training examples and then two negative. We plot how often GPT-3 2.7B predicts Positive on the balanced validation set. When the prompt is unbalanced, the predictions are unbalanced (*majority label bias*). In addition, balanced prompts that have one class repeated near the end, e.g., end with two Negative examples, will have a bias towards that class (*recency bias*).

Zhao et al. 2021. "Calibrate Before Use: Improving Few-Shot Performance of Language Models"

Problem 1: some labels are preferred than others

A boring story. It was ________

Language Model →

great	55%
terrible	45%

N/A. It was ________

Language Model →

great	75%
terrible	25%

Zhao et al. 2021. “Calibrate Before Use: Improving Few-Shot Performance of Language Models”
Holtzman et al 2021. "Surface Form Competition: Why the Highest Probability Answer Isn't Always Right"

Problem 1: some labels are preferred than others

A beautiful film. It was great.
A master class. It was great.
A boring story. It was ________

Language Model →

great	75%
terrible	25%

A beautiful film. It was great.
A master class. It was great.
N/A. It was ________

Language Model →

great	95%
terrible	5%

Zhao et al. 2021. “Calibrate Before Use: Improving Few-Shot Performance of Language Models”
Holtzman et al 2021. "Surface Form Competition: Why the Highest Probability Answer Isn't Always Right"

Problem 2: Surface form competition

A three-hour cinema master class. It was ________

Language Model →

great
awesome
excellent
fantastic
perfect
terrific
wonderful
exceptional

Zhao et al. 2021. “Calibrate Before Use: Improving Few-Shot Performance of Language Models”
Holtzman et al 2021. "Surface Form Competition: Why the Highest Probability Answer Isn't Always Right"

Problem 2: Surface form competition

A three-hour cinema master class. It was ________

Language Model →

great (3%)
awesome
excellent (90%)
fantastic
perfect
terrific
wonderful
exceptional

Zhao et al. 2021. “Calibrate Before Use: Improving Few-Shot Performance of Language Models”
Holtzman et al 2021. "Surface Form Competition: Why the Highest Probability Answer Isn't Always Right"

Solution 1: Calibrate model scores

Subpar acting.	It was terrible.
Beautiful film.	It was great.
A master class.	It was _________

→ P("terrible")
P("great")

Subpar acting.	It was terrible.
Beautiful film.	It was great.
N/A	It was _________

→ $P_{\_n/a}$("terrible")
$P_{\_n/a}$("great")

How much did a likelihood of ***terrible*** change?

How much did a likelihood of ***great*** change?

$$\log P_{\_final}(\text{“terrible”}) = \log P(\text{“terrible”}) - \log P_{\_n/a}(\text{“terrible”})$$
$$\log P_{\_final}(\text{“great”}) = \log P(\text{“great”}) - \log P_{\_n/a}(\text{“great”})$$

Zhao et al. 2021. “Calibrate Before Use: Improving Few-Shot Performance of Language Models”
Holtzman et al 2021. "Surface Form Competition: Why the Highest Probability Answer Isn't Always Right"

Solution 2: Noisy Channel

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \propto P(x|y)P(y)$$

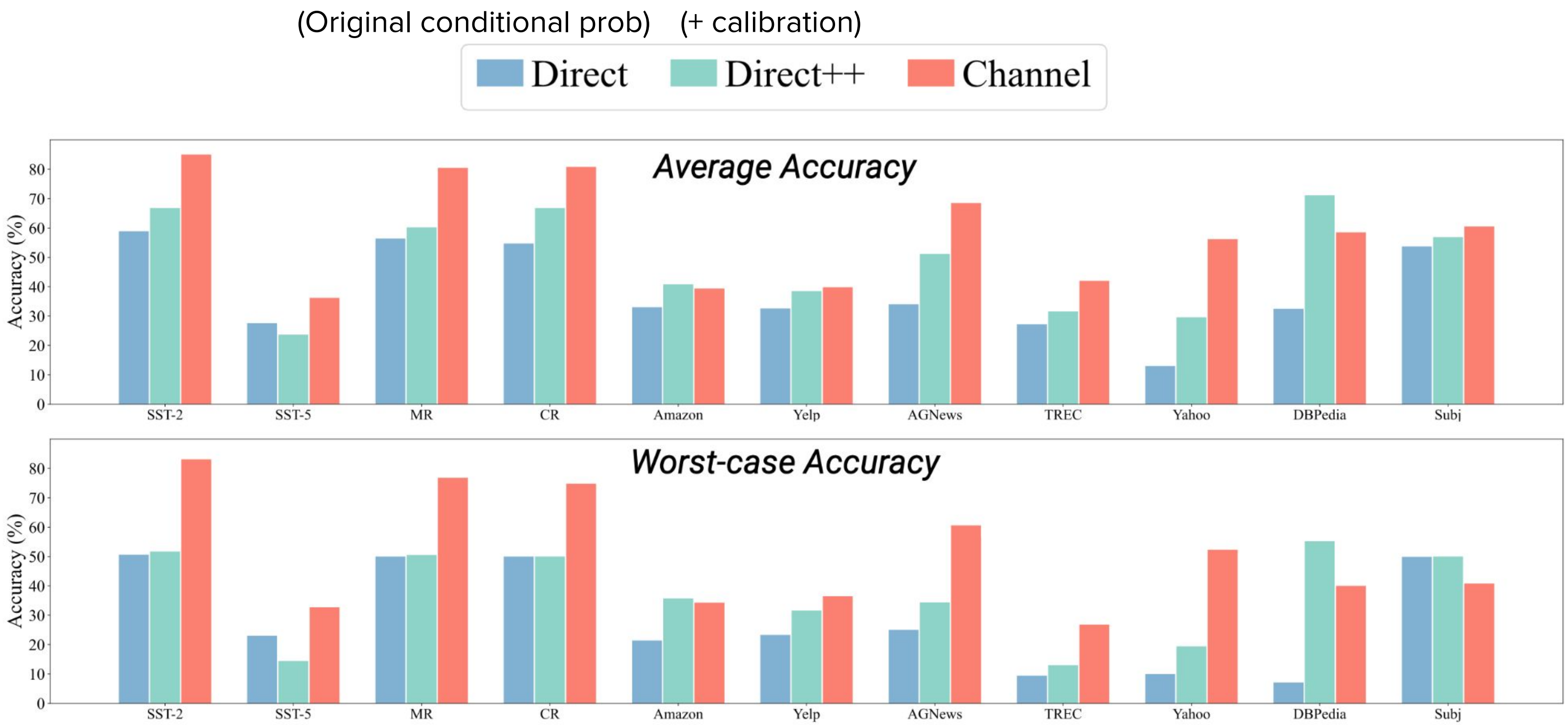
P(“**It was great**” | “A three-hour cinema master class.”)
P(“**It was terrible**” | “A three-hour cinema master class.”)

↓

P(“A three-hour cinema master class.” | “**It was great**”)
P(“A three-hour cinema master class.” | “**It was terrible**”)

Min et al. 2022. "Noisy Channel Language Model Prompting for Few-Shot Text Classification"

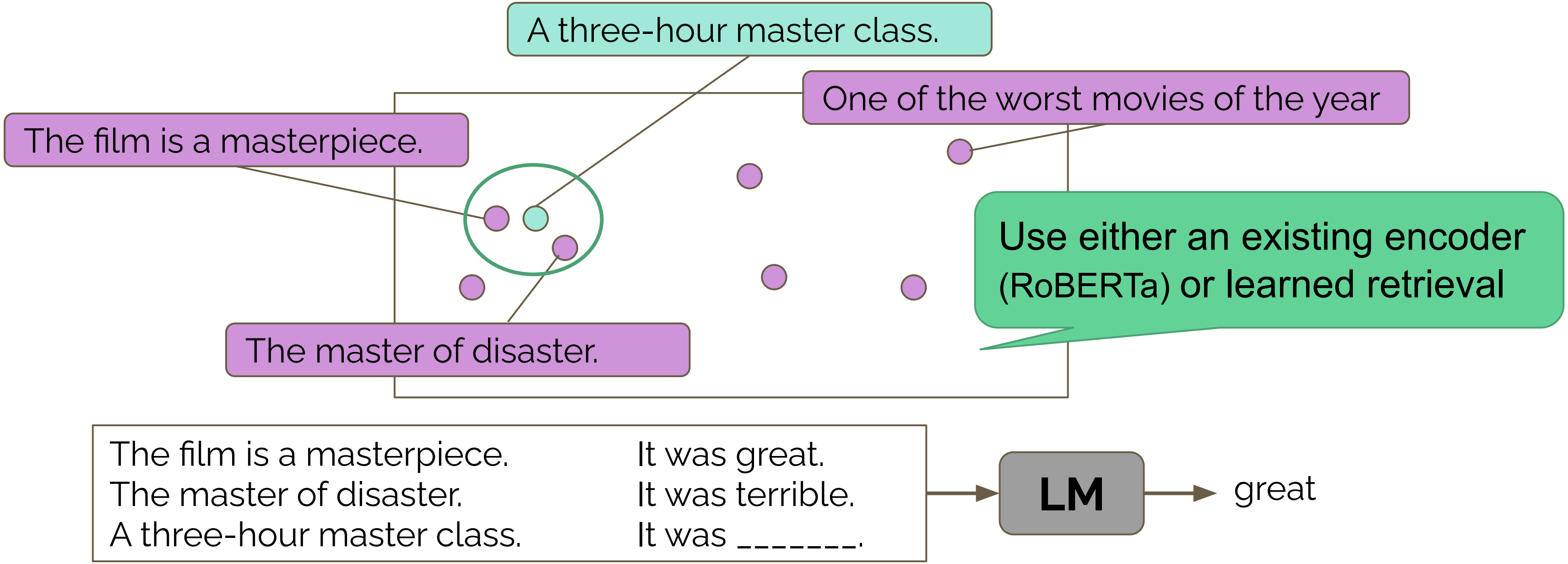
Solution 2: Noisy Channel



Min et al. 2022. "Noisy Channel Language Model Prompting for Few-Shot Text Classification"

How to choose the best *k* examples?

Assumption: you already have the labeled data that is large enough



Liu et al. 2021. What Makes Good In-Context Examples for GPT-3?
Rubin et al. 2021. "Learning To Retrieve Prompts for In-Context Learning"

How to order k examples?

Review: A mostly tired retread of several other mob tales.
Sentiment: negative

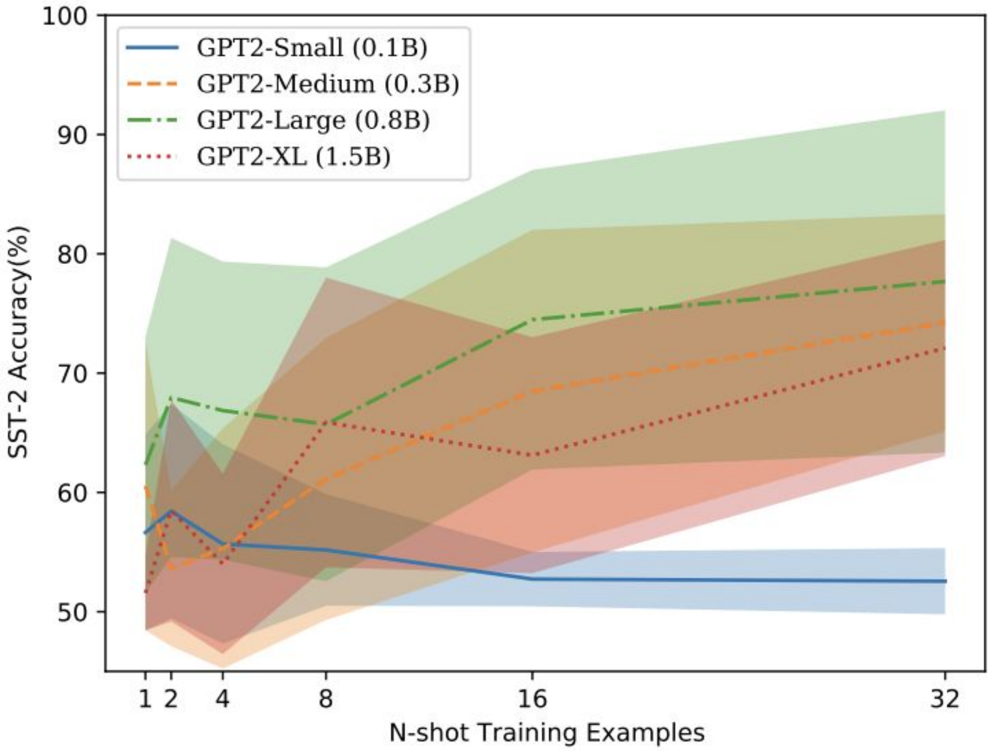
Review: The film is the masterpiece.
Sentiment: negative

Review: One of the worst movies of the year.
Sentiment: negative

Skipping the methodology, but is an important dimension of demonstrations!

Lu et al. 2022. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity”

How to order k examples?



Lu et al. 2022. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity”

How to order k examples?

Step 1: Generate **unlabeled dev set**

Step 2: **Score** each permutation based on unlabeled dev set

Step 3: **Choose** the best permutation!

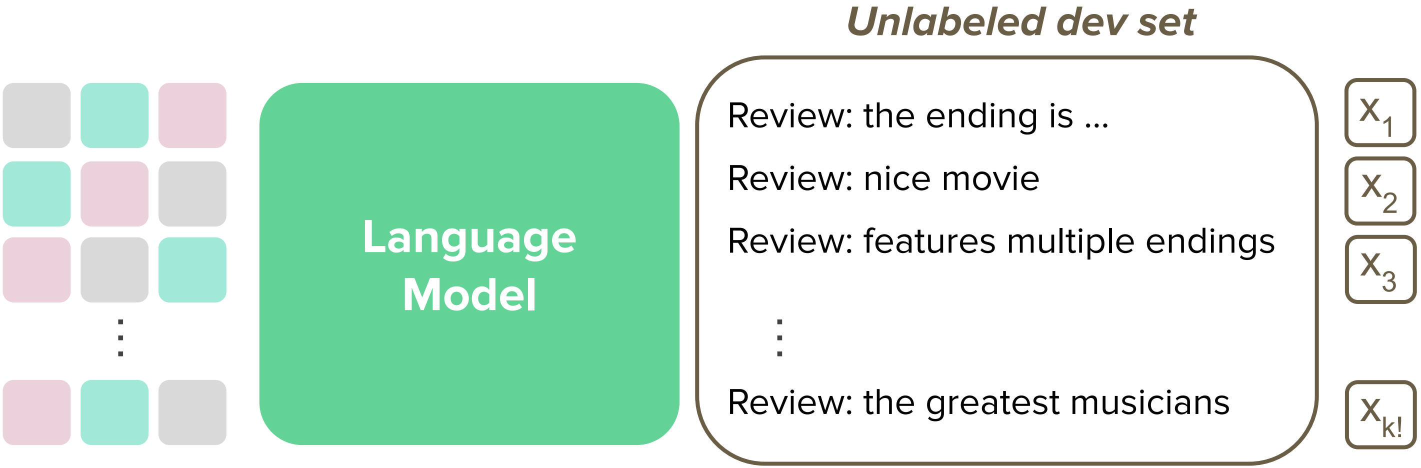
Lu et al. 2022. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity”

How to order k examples?

Step 1: Generate **unlabeled dev set**

Step 2: **Score** each permutation based on unlabeled dev set

Step 3: **Choose** the best permutation!

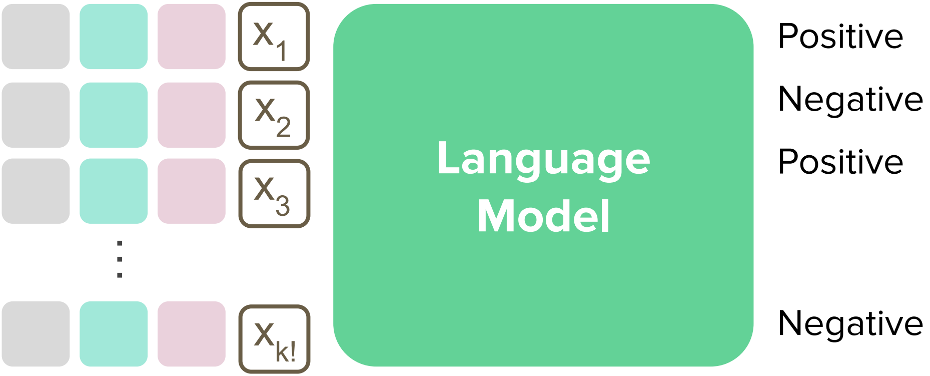


Lu et al. 2022. "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity"

How to order k examples?

1. GlobalE

Intuition: model prediction over *k!* examples should be evenly distributed



p_v: Portion of examples whose prediction is *v*

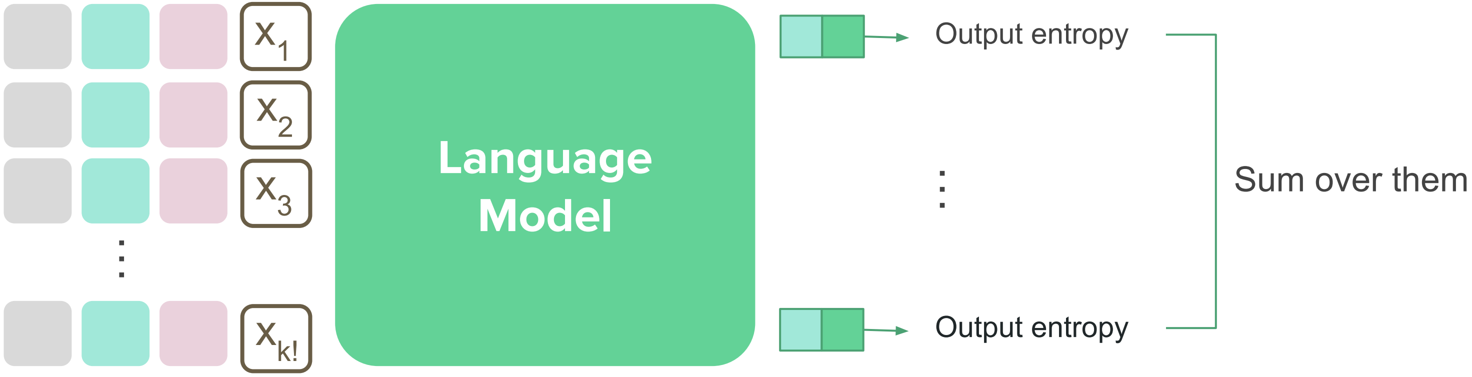
$$\text{GlobalE} = \Sigma_{v \in \mathcal{V}} \left(-p_v \log p_v\right)$$

Lu et al. 2022. “Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity”

How to order k examples?

2. LocalE

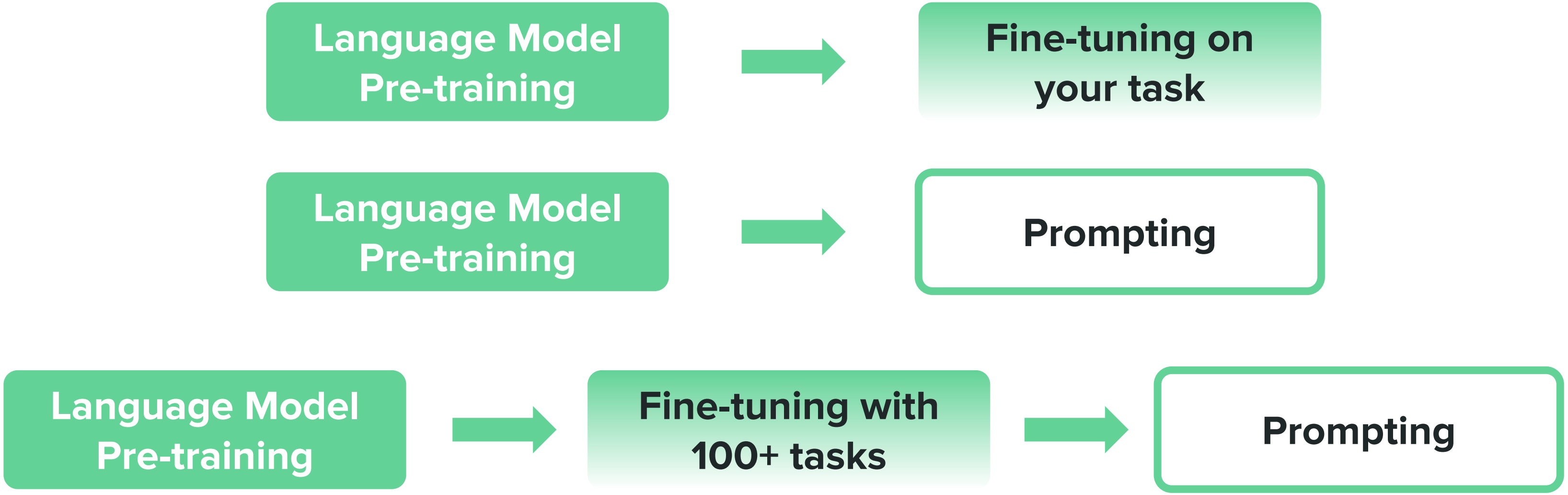
Intuition: model output shouldn't be overly confident



$$\text{LocalE} = \Sigma_{1 \leq i \leq k!} \Sigma_{v \in \mathcal{V}} \left(-P(v|x_i) \log P(v|x_i)\right)$$

Lu et al. 2022. "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity"

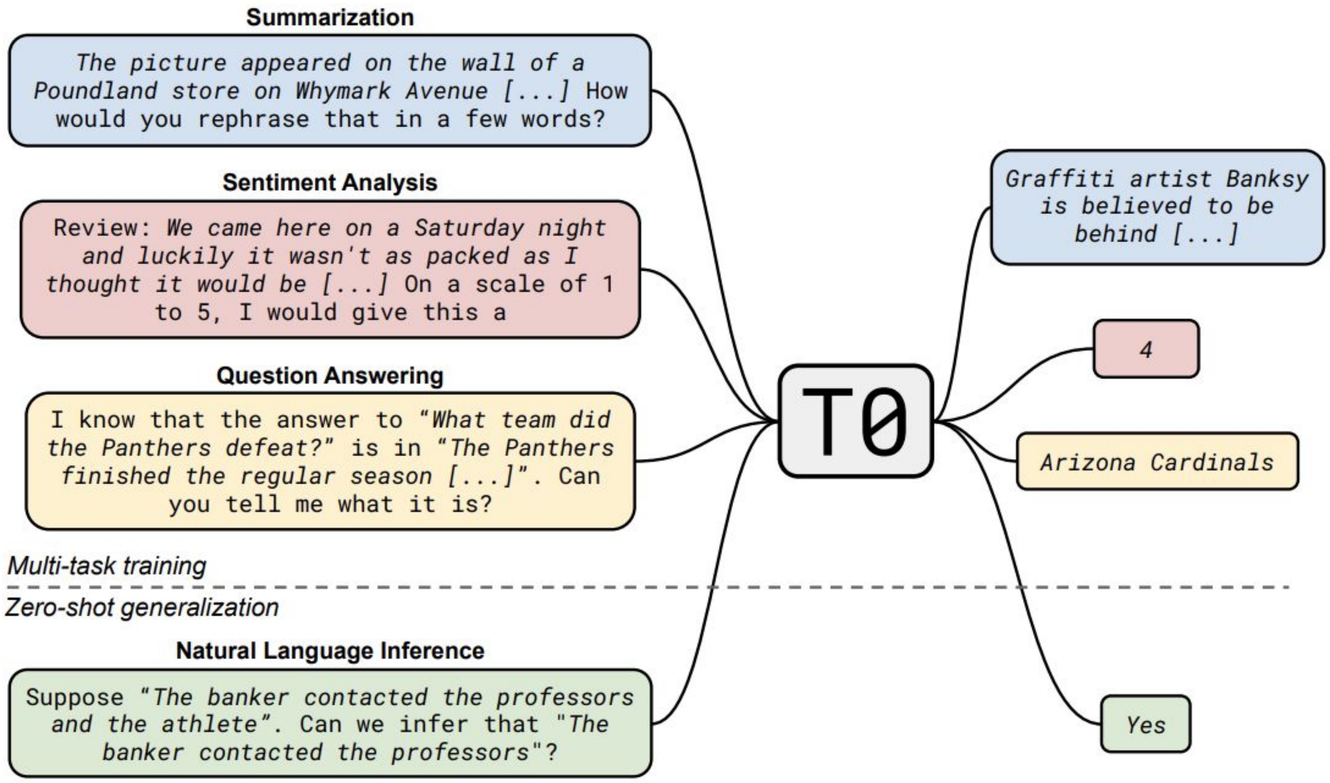
Multi-task learning for prompting



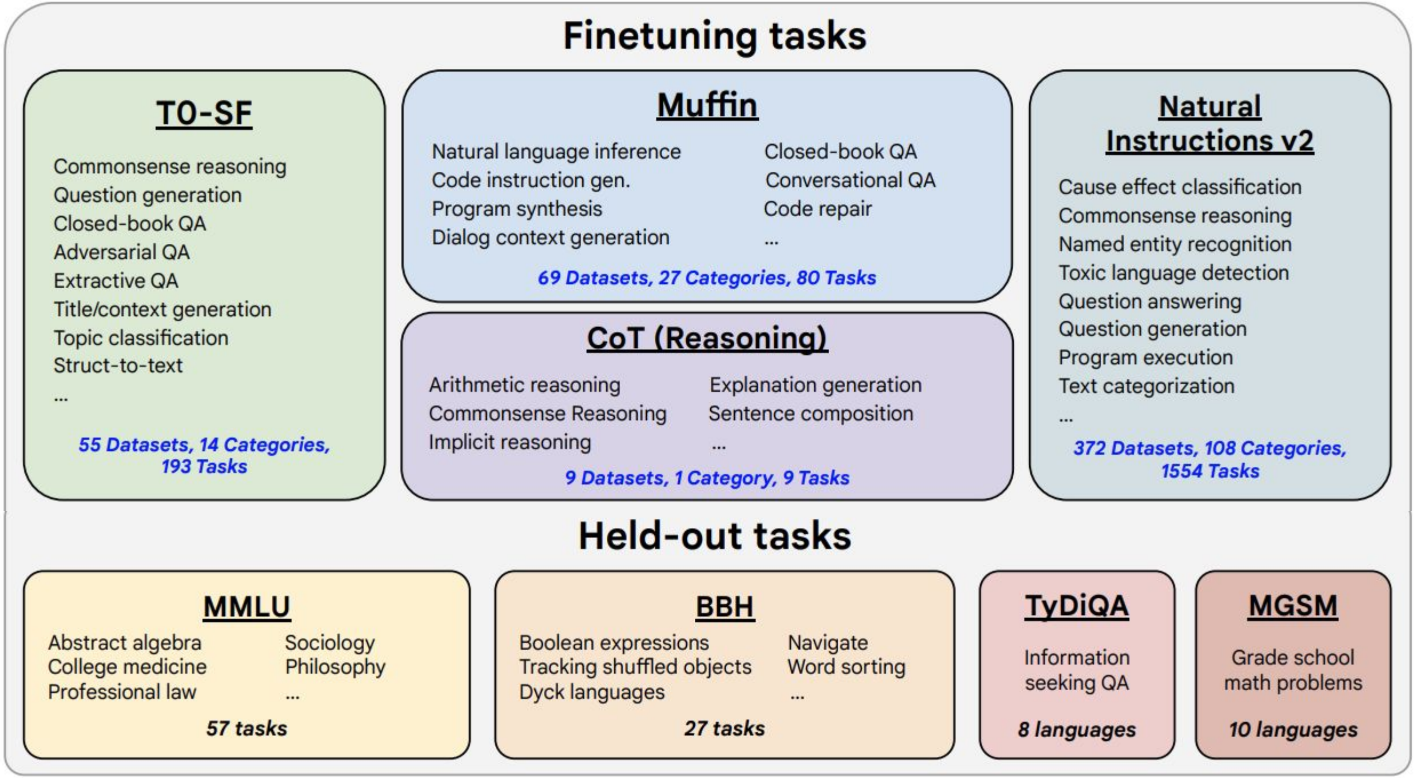
Remember, it’s **still prompting** (no fine-tuning on the target task)

Sanh et al., 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization
Wei et al., 2022. Finetuned Language Models Are Zero-Shot Learners
Min et al. 2022. MetaICL: Learning to Learn In Context
Wang et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks
Chung et al. 2022. Scaling Instruction-Finetuned Language Models

Multi-task learning for prompting



Multi-task learning for prompting



Multi-task learning for prompting (why?)

- Even though the pretrained language model works for prompting/in-context learning, it actually has never seen the format of prompting/in-context learning.
- **Simply exposing to the format of prompting/in-context learning could greatly improve performance.**
- We already have a large number of labeled datasets we've already collected, so why not use them?
- But if you'll fine-tune the model... why not fine-tune on the target task?
  - **Consider multi-task learning as part of pretraining**
  - You do multi-task learning ***only once***, and can use this model ***frozen*** for ***any new*** downstream task.

Multi-task learning for prompting (method)

Given (x, y), maximize $P_{\mathrm{LM}}(v(y)|f(x))$ where...

- (x, y) is from a dataset sampled from a large collection of datasets
- f and v are sampled from a collection of different formats

  - Prompt to include in-context examples

    An effortlessly accomplished and richly resonant work. It was great!
    A mostly tired retread of several other mob tales. It was terrible!

  - Prompt to include natural language description about the task

    "Identify the sentiment of this movie review."

  - Verbalizer to include rationale about the output

Question: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left?

Answer: 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.

Chung et al. 2022. Scaling Instruction-Finetuned Language Models

Review

- LM prompting & In-context learning show promising results, but their performance is highly unstable/brittle
- Better scoring
  - Calibration
  - Noisy Channel
- Better formation of demonstrations
  - Better choice of in-context examples
  - Better permutations of in-context examples
- Multi-task learning for prompting

True Few-Shot Learning

“We are unconsciously cheating on the data, and few-shot performance is overestimated”

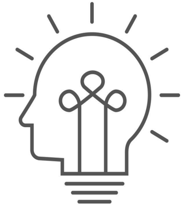
- Use of large development data
- Choice of patterns and verbalizers
- Choice of various hyperparameters

Should be careful in evaluation

Perez et al 2021. "True Few-Shot Learning with Language Models"

How/Why in-context learning works?

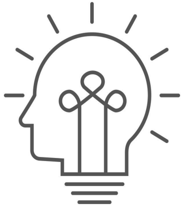
Transition: How/Why in-context learning works?



Any arbitrary task ⇨ **Language Model**

A few-shot learner

Transition: How/Why in-context learning works?



Any arbitrary task

Language Model

A few-shot learner

How/Why in-context learning works?

- Demonstrations do not teach a new task; instead, it is about locating an already-learned task during pretraining (Reynolds & McDonell, 2021)
- LMs do not exactly understand the meaning of their prompt (Webson & Pavlick, 2021)
- Demonstrations are about providing a latent concept so that LM generates coherent next tokens (Xie et al. 2022)
- In-context learning performance is highly correlated with term frequencies during pretraining (Razeghi et al. 2022)
- LMs do not need input-label mapping in demonstrations, instead, it uses the specification of the input & label distribution separately (Min et al. 2022)
- Data properties lead to the emergence of few-shot learning (burstiness, long-tailedness, many-to-one or one-to-many mappings, a Zipfian distribution) (Chan et al. 2022)

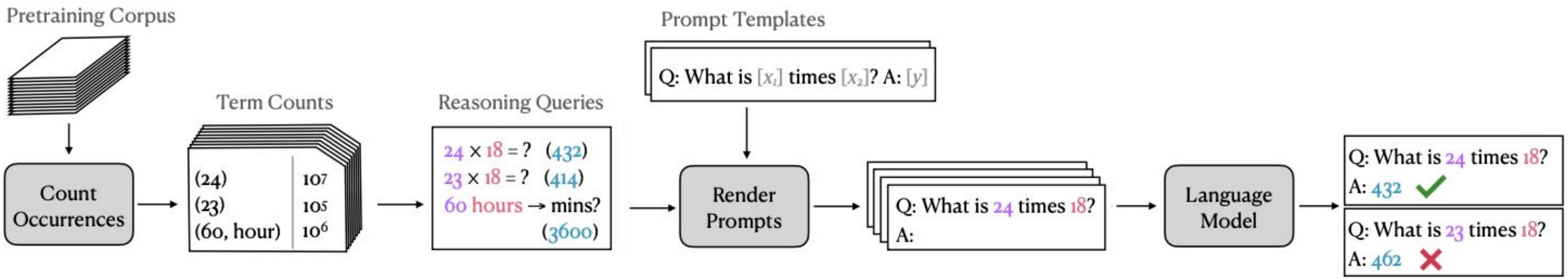
How/Why in-context learning works?

- Demonstrations do not teach a new task; instead, it is about locating an already-learned task during pretraining (Reynolds & McDonell, 2021)
- LMs do not exactly understand the meaning of their prompt (Webson & Pavlick, 2021)
- Demonstrations are about providing a latent concept so that LM generates coherent next tokens (Xie et al. 2022)
- **In-context learning performance is highly correlated with term frequencies during pretraining** (Razeghi et al. 2022)
- **LMs do not need input-label mapping in demonstrations, instead, it uses the specification of the input & label distribution separately** (Min et al. 2022)
- Data properties lead to the emergence of few-shot learning (burstiness, long-tailedness, many-to-one or one-to-many mappings, a Zipfian distribution) (Chan et al. 2022)

Impact of Pretraining Term Frequencies

In-context learning performance is highly correlated with term frequencies during pretraining

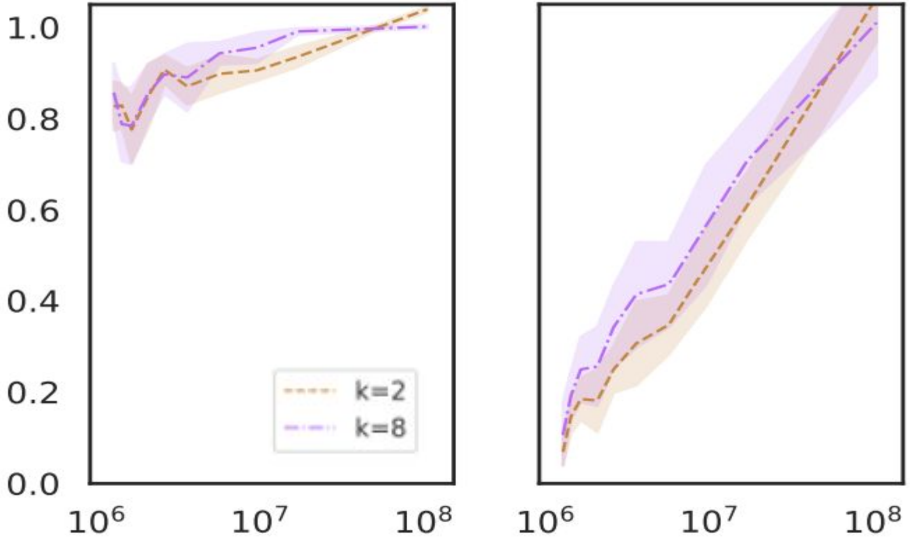
- For each task, identify relevant terms from each instance—numbers and units
- Count co-occurrences of these terms in the pretraining data (term pairs or triples within a fixed window)



Razeghi et al. 2022. "Impact of Pretraining Term Frequencies on Few-Shot Reasoning"

Impact of Pretraining Term Frequencies

In-context learning performance is highly correlated with term frequencies during pretraining

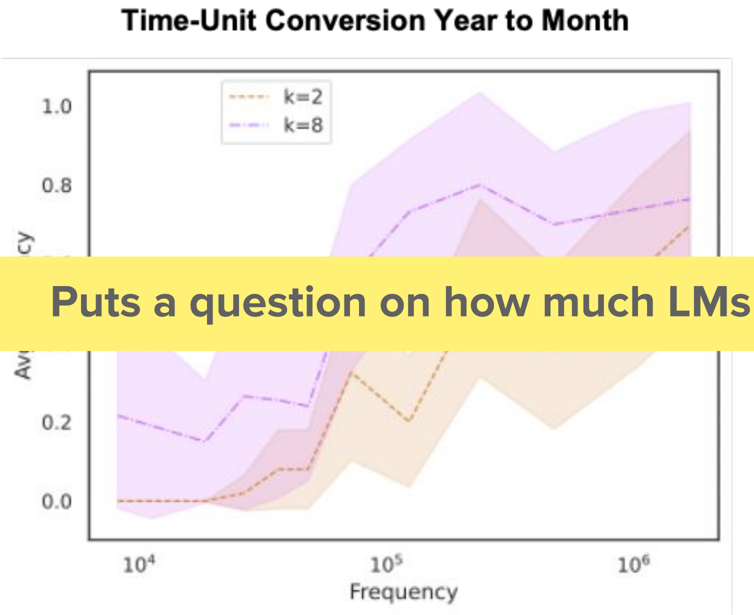


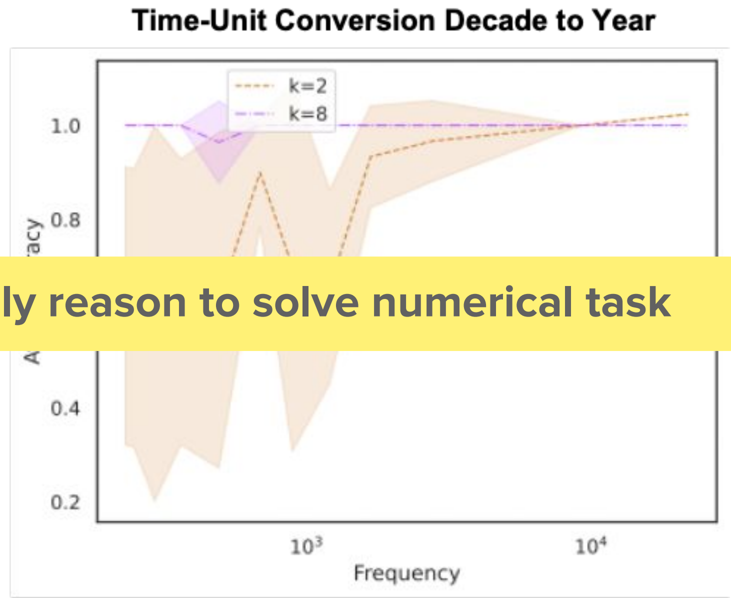
(a) Arithmetic-Addition (b) Arithmetic-Multiplication

Razeghi et al. 2022. "Impact of Pretraining Term Frequencies on Few-Shot Reasoning"

Impact of Pretraining Term Frequencies

In-context learning performance is highly correlated with term frequencies during pretraining





Puts a question on how much LMs actually reason to solve numerical task

Razeghi et al. 2022. "Impact of Pretraining Term Frequencies on Few-Shot Reasoning"

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

Input: A three-hour master class.
Label: ________

Language Model

Input: An effortlessly accomplished and richly resonant work.
Label: **negative**

Input: A mostly tired retread of several other mob tales.
Label: **positive**

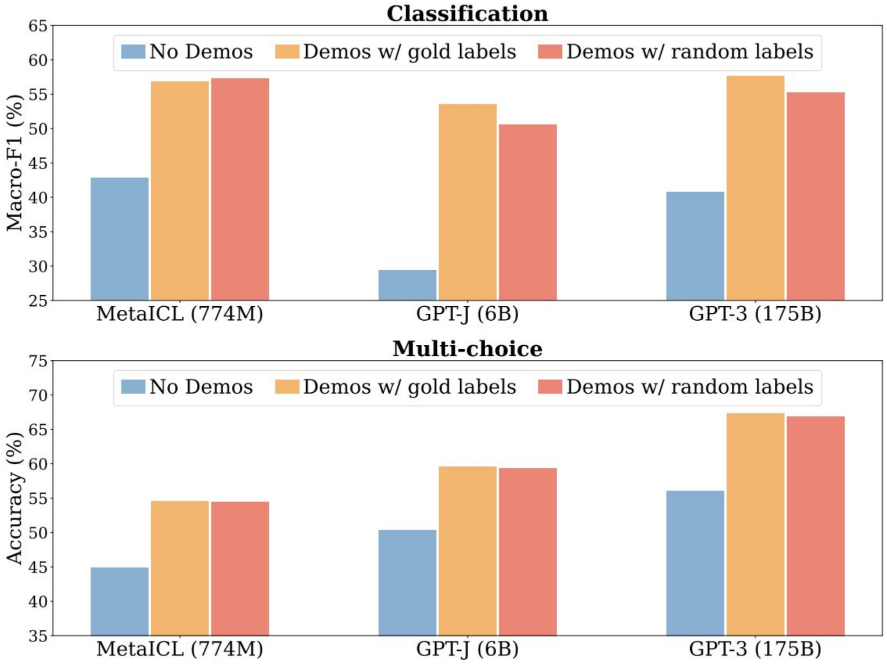
Input: A three-hour master class.
Label: ________

Language Model

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

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping



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

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

Input: A three-hour master class.
Label: _______

Language Model

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

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: Colour-printed lithograph. Very good condition.
Label: positive

Input: Many accompanying marketing … meaning.
Label: negative

Input: A three-hour master class.
Label: _______

Language Model

Removing correct input distribution significantly drops performance

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

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: An effortlessly accomplished and richly resonant work.
Label: Unanimity

Input: A mostly tired retread of several other mob tales.
Label: Wave

Input: A three-hour master class.
Label: _______

Language Model

Removing correct input distribution significantly drops performance

Removing correct label space significantly drops performance

Input and label distributions matter *independently*

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

How about non-classification?

Demonstration

Question: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left?

Answer: 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.

Test input

Question: 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. How many pages are left?

Answer: ________________

Language Model

Model output

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. The answer is 84.

Madaan & Yazdanbakhsh, 2022. Text and Patterns: For Effective Chain of Thought, It Takes Two to Tango
Wang et al. 2022. "Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters"

How about non-classification?

Question: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left?

Answer: 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, they have 10 + 35 = 45 in total. The answer is 45.

Question: 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. How many pages are left?

Answer: ______________

Language Model

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. The answer is 84.

The correctness of in-context examples is not necessary.

Madaan & Yazdanbakhsh, 2022. Text and Patterns: For Effective Chain of Thought, It Takes Two to Tango
Wang et al. 2022. “Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters”

Takeaways

- Does language model magically learn a new task **as defined in** the demonstration? **Maybe not!**
- Findings 1) Accuracy depends a lot on how many times relevant terms appear in the pretraining data
- Findings 2) Even if demonstrations provided to the language model are incorrect, the model still performs the original task well
- These suggest that in-context learning might be mainly about **recovering the task that is implicitly learned during pre-training**, using the demonstrations as semantic cues.
- This is an ongoing topic of debate and active research!

Summary

Summary & Open questions

- Prompting/In-context learning
  - No need for gradient updates ➜ Much easier to use large models!
- Better calibration, better scoring of model outputs, better formation of demonstrations & multi-task learning lead to great improvements
  - How to make it less sensitive?
  - It increases inference cost – how to make it efficient?
  - How to scale it (longer context, more training examples, wider range of tasks)?
- Need to be cautious in evaluation
- Still in progress on understanding how/why it works, with papers showing that in-context learning is about *task location* rather than learning a *new* task
  - Can we predict whether in-context learning would work on a given task or not?

Reminding the timeline

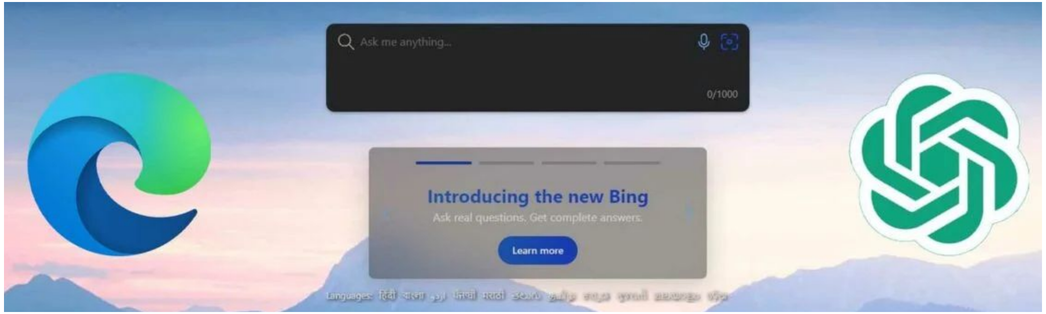
- Before 2018: Supervised training with LSTM/etc...
- 2018: Advent of Pretrained LMs + Fine-tuning
- 2020: The GPT-3 paper introduces Prompting and In-Context Learning
- 2021: Much work about how to improve them
- 2022:
  - Multi-task learning for prompting
  - Understanding prompting and in-context learning
- 2023: ?

Things we didn’t cover

- Multi-task learning with human feedback
- Using language models for various applications
- ChatGPT (!!)

Things we didn’t cover

- Multi-task learning with human feedback
- Using language models for various applications
- ChatGPT (!!)
- Retrieval (Search) + language modeling



Google shares tank 8% as AI chatbot Bard flubs answer in ad

Shares of Google’s parent company lost more than $100bn after its Bard chatbot advertisement showed inaccurate information.

Questions?

Useful resources

- ACL 2022 Tutorial in Zero-/Few-shot learning with Pretrained Language Models
- Princeton class in Understanding Large Language Models
- Johns Hopkins class on Self-supervised Statistical Models
- An interview with Sameer Singh on ChatGPT, GPT-4 and Cutting Edge Research
- Stanford Blog Post on How In-Context Learning Works