CSCE 670 - Information Storage and Retrieval

Lecture 21: Large Language Models Basics

Yu Zhang

yuzhang@tamu.edu

November 6, 2025

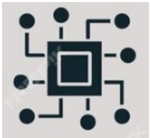
Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html

We are finally here!

- ✅ Phase 1: Search Engines
  - basics, Boolean and ranked retrieval, link analysis, evaluation, learning to rank (ML + ranking), …
- ✅ Phase 2: Recommender Systems
  - basics, non-personalized recommendation, collaborative filtering, matrix factorization, implicit recommendation, …
- ✅ Phase 3: From Foundations to Modern Methods
  - embedding learning, Transformer, "small" language models, … (for search and recommendation)
- Phase 4: Large Language Models (!!)

NLP before the Era of Large Language Models

- Given an NLP task
  - Step 1: Find/annotate task-specific training data (e.g., 10,000 training samples)
  - Step 2: Design a task-specific model (e.g., 3-layer recurrent neural networks with …)
  - Step 3: Use the annotated data to train the model

Math Word Problem: (*John buys 20 cards and 1/4 are uncommon. How many uncommon cards did he get?*, *5*) → Math Model 

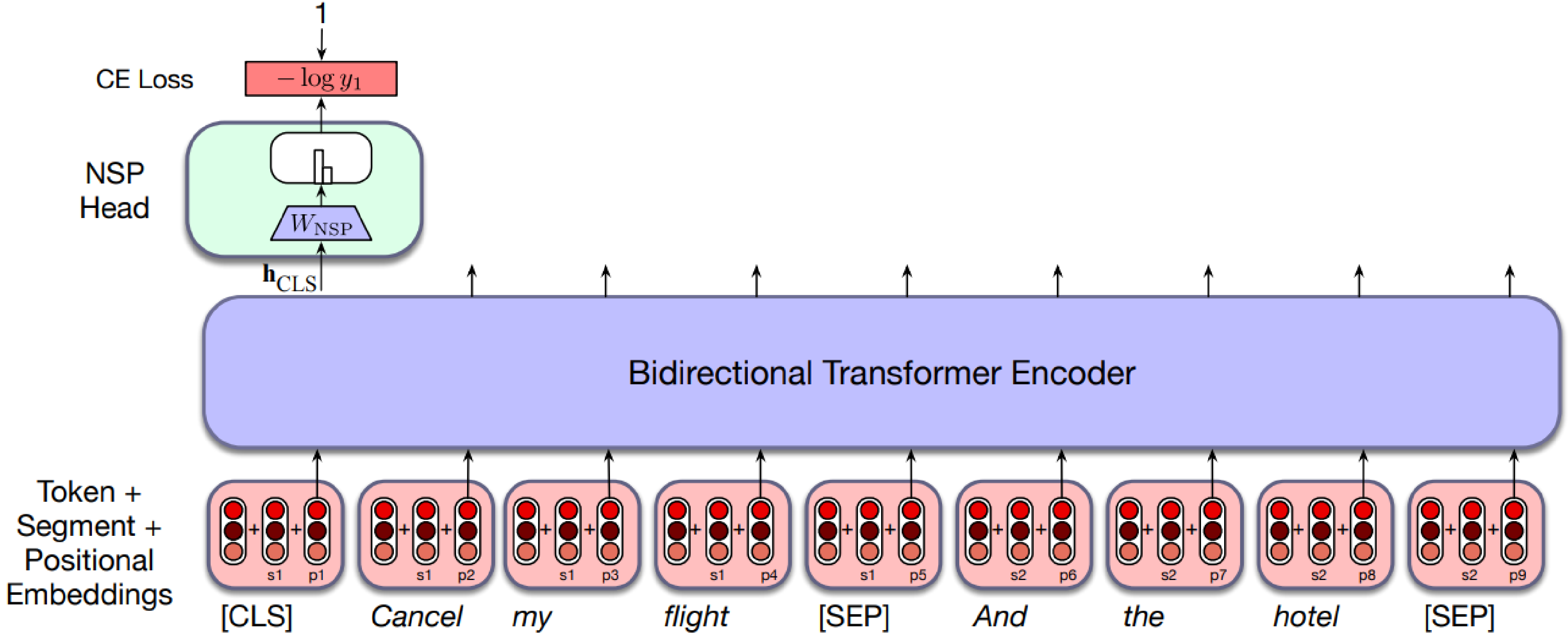
Paper Classification: (*Rapid chromatographic technique for preparative separations with moderate resolution*, *Organic Chemistry*) → Classification Model 

Information Extraction: (*in rats, nitrofurantoin causes pulmonary toxicity*, *[pulmonary toxicity, DISEASE]*) → IE Model 

Question Answering: (*Who formulated the zeroth law of thermodynamics?*, *Ralph H. Fowler*) → QA Model 

Recap: BERT [Devlin et al., NAACL 2019]

- BERT has already learned knowledge, linguistic patterns, and other generally useful information for various NLP tasks from the entire Wikipedia and BookCorpus.
- However, it cannot be directly used for many downstream tasks without any further fine-tuning.



How to fine-tune BERT for different NLP tasks?

- *Math Word Problem*: Input the entire math word problem into BERT, get the [CLS] token embedding, and perform regression by training a regression layer
- *Paper Classification*: Input the paper into BERT, get the [CLS] token embedding, and perform topic classification by training a classification layer
- *Information Extraction*: Input the text into BERT, get the embedding of each token, and classify each token

in	*rats*	*,*	*nitrofurantoin*	*causes*	*pulmonary*	*toxicity*
None	*None*	*None*	*B-Chemical*	*None*	*B-Disease*	*I-Disease*

- Suppose you train a math word problem solver from scratch (with all parameters initialized randomly), it might require around 10,000 training samples to achieve good performance.
- In contrast, fine-tuning BERT typically needs far fewer training samples (for example, around 100) to reach comparable performance.

The BERT Revolution

foundation model

100 *Math Word Problem* training samples

100 *Question Answering* training samples

100 *Paper Classification* training samples

100 *Information Extraction* training samples

Math Word Problem: (*John buys 20 cards and 1/4 are uncommon. How many uncommon cards did he get?*, *5*)

Paper Classification: (*Rapid chromatographic technique for preparative separations with moderate resolution*, *Organic Chemistry*)

Information Extraction: (*in rats, nitrofurantoin causes pulmonary toxicity*, *[pulmonary toxicity, DISEASE]*)

Question Answering: (*Who formulated the zeroth law of thermodynamics?*, *Ralph H. Fowler*)

But today's ChatGPT is far more powerful!

Solve the following Math Word Problem:
John buys 20 cards and 1/4 are uncommon. How many uncommon cards did he get?

0 *Math Word Problem* training samples

5

Classify the following paper. Candidate classes are [XXX, XXX, …]:
Rapid chromatographic technique for preparative separations with …

0 *Paper Classification* training samples

Organic Chemistry

Find all the DISEASE entities in the following sentence:
in rats, nitrofurantoin causes pulmonary toxicity

0 *Information Extraction* training samples

pulmonary toxicity

Answer the following question:
Who formulated the zeroth law of thermodynamics?

0 *Question Answering* training samples

Ralph H. Fowler

One Model for All Tasks!

Let's take a step back first

The task instruction is optional if you have 5 training samples.

5 *Math Word Problem* training samples

John buys 20 cards and 1/4 are uncommon. How many uncommon cards did he get?

5 *Paper Classification* training samples

Rapid chromatographic technique for preparative separations with moderate resolution

5 *Information Extraction* training samples

in rats, nitrofurantoin causes pulmonary toxicity

5 *Question Answering* training samples

Who formulated the zeroth law of thermodynamics?

One Model for All Tasks!

5

Organic Chemistry

pulmonary toxicity

Ralph H. Fowler

GPT-3 [Brown et al., NeurIPS 2020]

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. Z

Christopher Hesse

Benjamin Cl

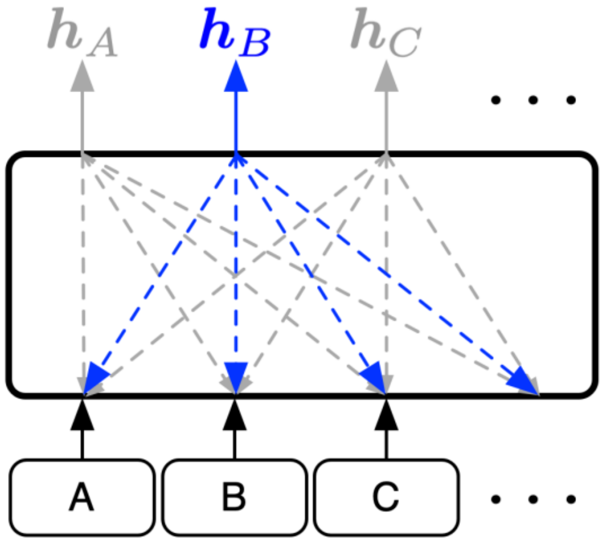
Language models are few-shot learners [PDF] r

T Brown, B Mann, N Ryder... - Advances in neural ..., 2020 - proceedings.neurips.cc

... up **language models** greatly improves task-agnostic, **few-shot** ... GPT-3, an autoregressive **language model** with 175 billion ... **language model**, and test its performance in the **few-shot** ...

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

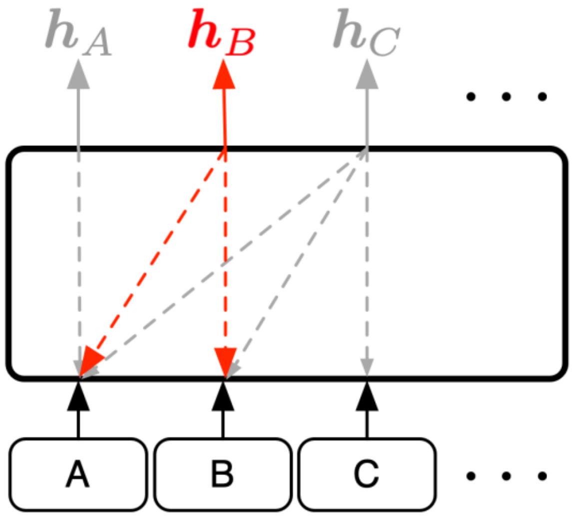
Recap: Two Types of Transformer Architecture



every token attends to all tokens

Bidirectional Self-Attention

q1·k1	q1·k2	q1·k3	q1·k4
q2·k1	q2·k2	q2·k3	q2·k4
q3·k1	q3·k2	q3·k3	q3·k4
q4·k1	q4·k2	q4·k3	q4·k4



every token attends to its previous tokens

Unidirectional Self-Attention

q1·k1	$-\infty$	$-\infty$	$-\infty$
q2·k1	q2·k2	$-\infty$	$-\infty$
q3·k1	q3·k2	q3·k3	$-\infty$
q4·k1	q4·k2	q4·k3	q4·k4

What if we pre-train a Transformer decoder?

- With only one task – next token prediction – on web-scale corpora

$$\sum_i p(\underbrace{w_i}_{\text{next token}} \mid \underbrace{w_1, w_2, \ldots, w_{i-1}}_{\text{previous tokens}} ; \underbrace{\Theta}_{\text{model parameters}})$$

- Most NLP tasks can "reduce" to next token prediction.
  - *Math*: *{Input to the model}* John buys 20 cards and 1/4 are uncommon. How many uncommon cards did he get? The answer is
    - *{Model predicts the next token(s)}* 11
  - *Classification*: *{Input to the model}* (paper title) Rapid chromatographic technique for preparative separations with moderate resolution => (label)
    - *{Model predicts the next token(s)}* Organic Chemistry

What if we pre-train a Transformer decoder?

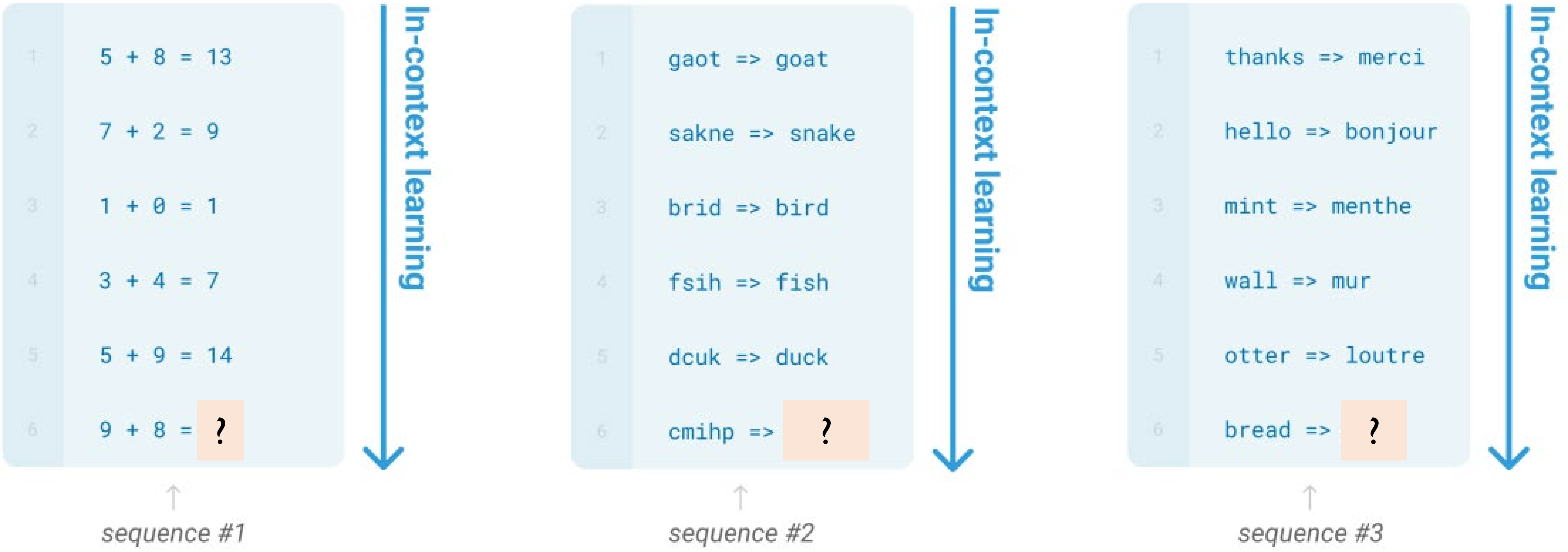
- Only one task – next token prediction – on web-scale corpora

$$\sum_i p(\underbrace{w_i}_{\text{next token}} \mid \underbrace{w_1, w_2, \dots, w_{i-1}}_{\text{previous tokens}} ; \underbrace{\Theta}_{\text{model parameters}})$$

- Most NLP tasks can "reduce" to next token prediction.
  - *Information Extraction*: {*Input to the model*} (text) in rats, nitrofurantoin causes pulmonary toxicity. => (entity, type)
    - {*Model predicts the next token(s)*} pulmonary toxicity, disease
  - *Question Answering*: {*Input to the model*} Who formulated the zeroth law of thermodynamics?
    - {*Model predicts the next token(s)*} Ralph H. Fowler

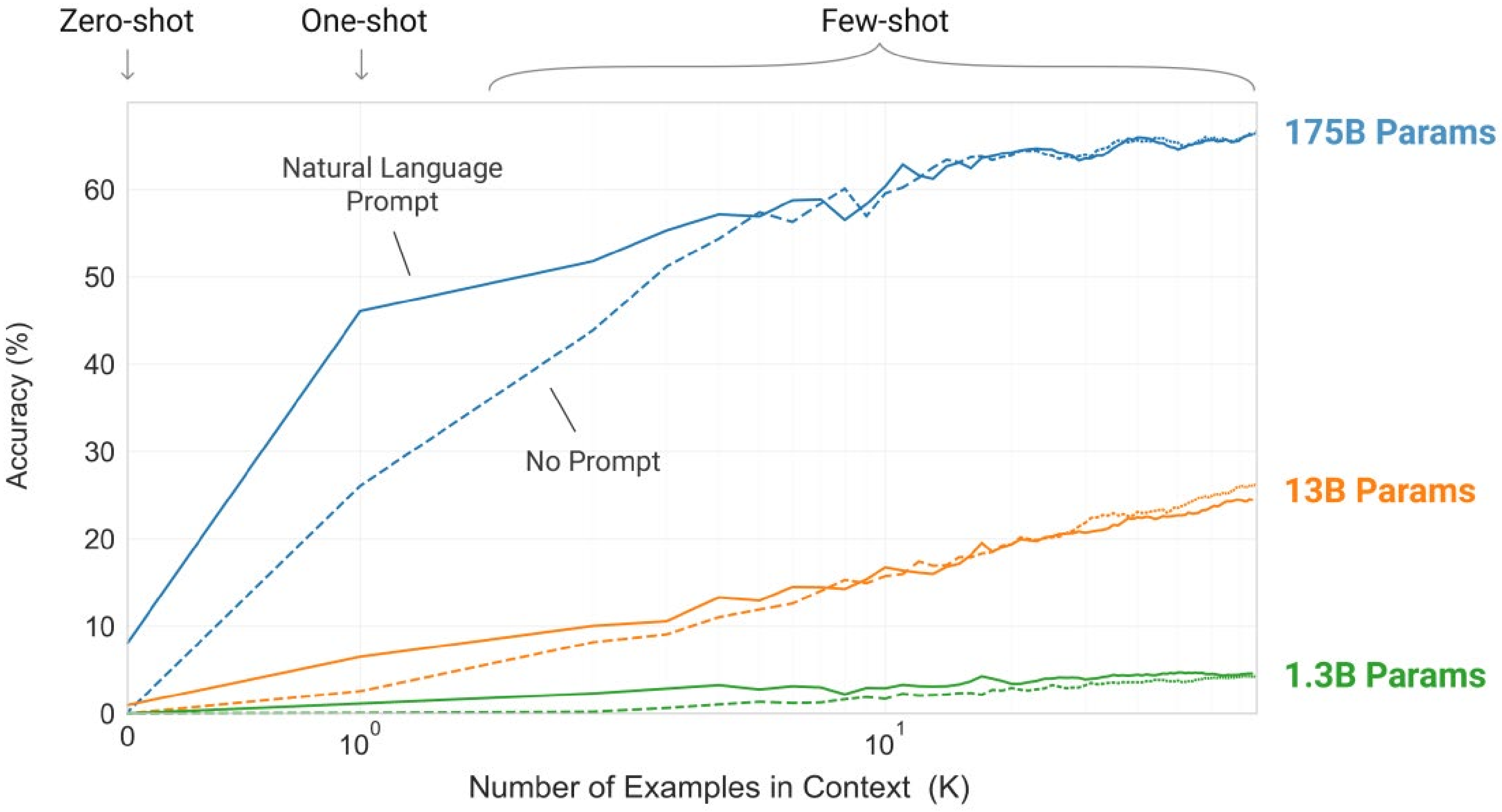
GPT-3 can performing each task with just a few examples

- The model may acquire a broad set of skills and pattern recognition abilities during pre-training. It then uses these abilities at inference time to rapidly adapt to or recognize the desired task. – "In-context learning"



Can a model be that "smart"?

- Only if it is big enough!



GPT-3

BERT-Base: 0.11B
BERT-Large: 0.34B

Can we make our demo examples more insightful?
(Prompt Engineering)

Chain-of-Thought Prompting [Wei et al., NeurIPS 2022]

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

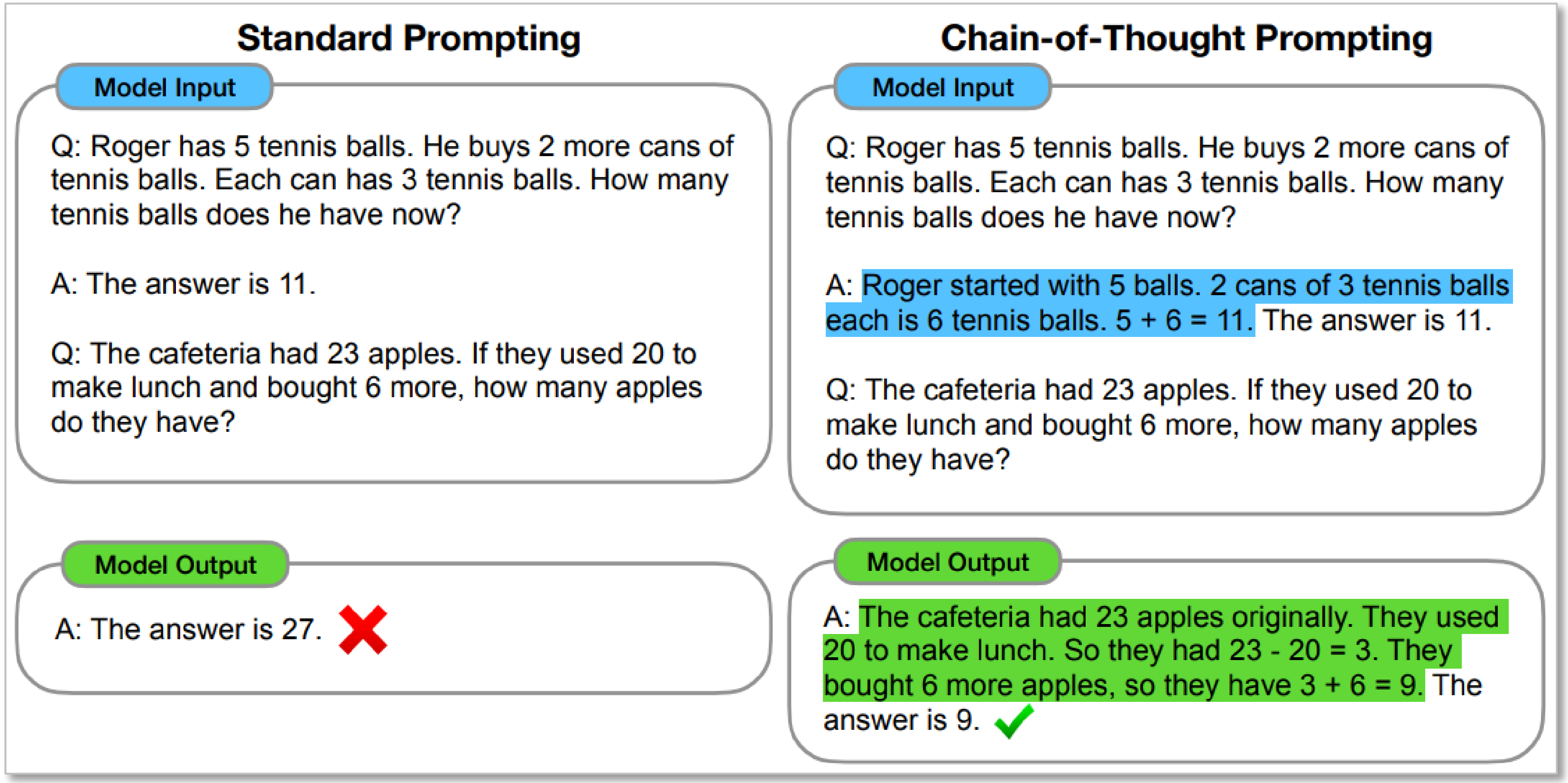
Jason Wei **Xuezhi Wang** **Dale Schuurmans** **Maarten Bosma**
Brian Ichter **Fei Xia** **Ed H. Chi** **Quoc V. Le** **Denny Zhou**

Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

We explore how gene
steps—significantly
complex reasoning.
naturally in sufficientl
thought prompting, where a few chain of thought demonstrations are provided as exemplars in prompting.

Chain-of-Thought Prompting

- Add a series of intermediate reasoning steps in the demonstration examples(s)
- Get the model to explain its reasoning steps before making an answer



Self-Consistency [Wang et al., ICLR 2023]

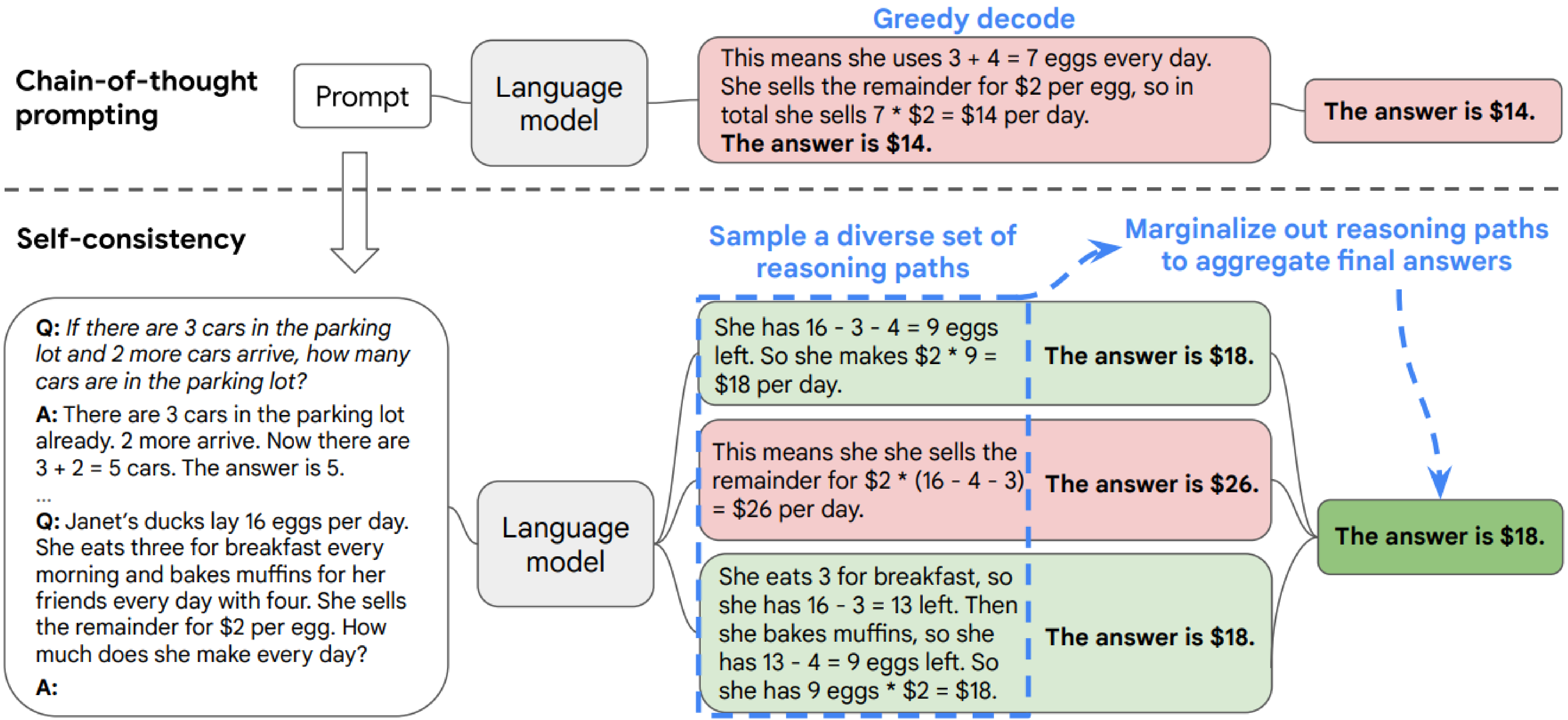
Self-Consistency Improves Chain of Thought Reasoning in Language Models

Xuezhi Wang[†‡] **Jason Wei**[†] **Dale Schuurmans**[†] **Quoc Le**[†] **Ed H. Chi**[†]
Sharan Narang[†] **Aakanksha Chowdhery**[†] **Denny Zhou**[†§]
[†]Google Research, Brain Team
[‡]xuezhiw@google.com, [§]dennyzhou@google.com

Abstract

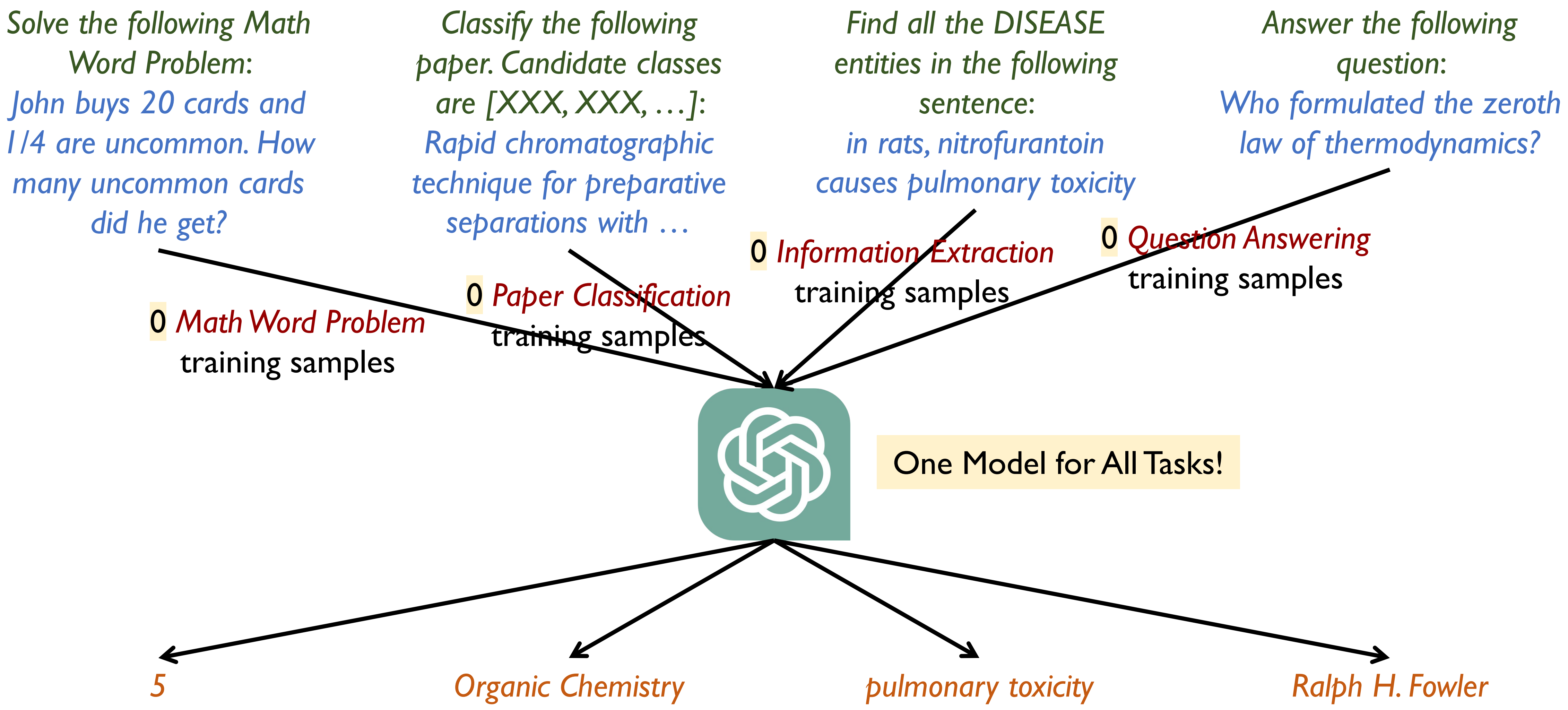
Chain-of-thought prompting combined with pre-trained large language models has achieved encouraging results on complex reasoning tasks. In this paper, we propose a new decoding strategy, *self-consistency*, to replace the naive greedy decoding used in chain-of-thought prompting. It first samples a diverse set of reasoning paths instead of only taking the greedy one, and then selects the most consistent answer by marginalizing out the sampled reasoning paths. Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer. Our extensive empirical evaluation shows that self-consistency boosts the performance of chain-of-thought prompting with a striking margin on a range of popular arithmetic and commonsense reasoning benchmarks, including GSM8K (+17.9%), SVAMP (+11.0%), AQuA (+12.2%), StrategyQA (+6.4%) and ARC-challenge (+3.9%).

Self-Consistency



From Few-Shot to Zero-Shot (Instruction Tuning and Alignment)

Our Final Goal



From Few-Shot to Zero-Shot

Few-shot

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

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

task description

examples

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 =>
```

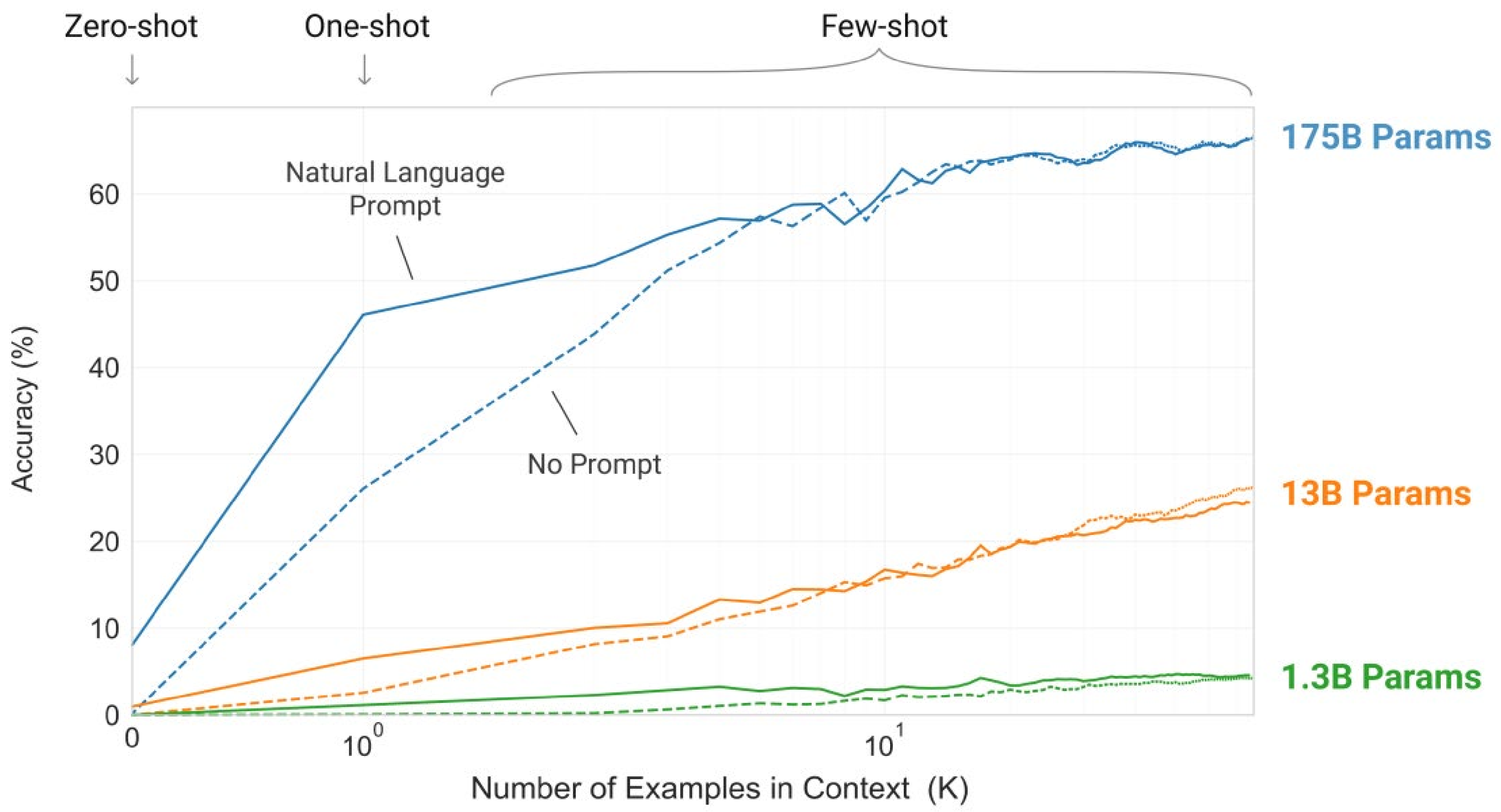
task description

prompt

Is GPT-3 a zero-shot learner?

Task Instruction Only

Task Instruction + A Few Examples



Why is the zero-shot setting hard?

- GPT-3 is not good at following an instruction to perform a new task.
  - Because it is never asked to do so during pre-training.
- How to solve this problem?
  - Tune the model to follow task instructions!

$$\sum_i p(\text{ ground truth output } \mid \text{ instruction, task input })$$

- How can we ensure that the model can handle unseen task instructions when it is actually used?
  - Let it see a sufficiently diverse set of task instructions during training so that it can generalize well
  - What really matters is the number of different tasks, not how many samples each task has
    - Find all available NLP benchmark datasets and convert them into the (instruction, input, output) format
    - Think of lots of creative new tasks and get people to label them

FLAN [Wei et al., ICLR 2022]

Finetuned Language Models Are Zero-Shot Learners

Jason Wei*, Maarten Bosma*, Vincent Y. Zhao*, Kelvin Guu*, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le

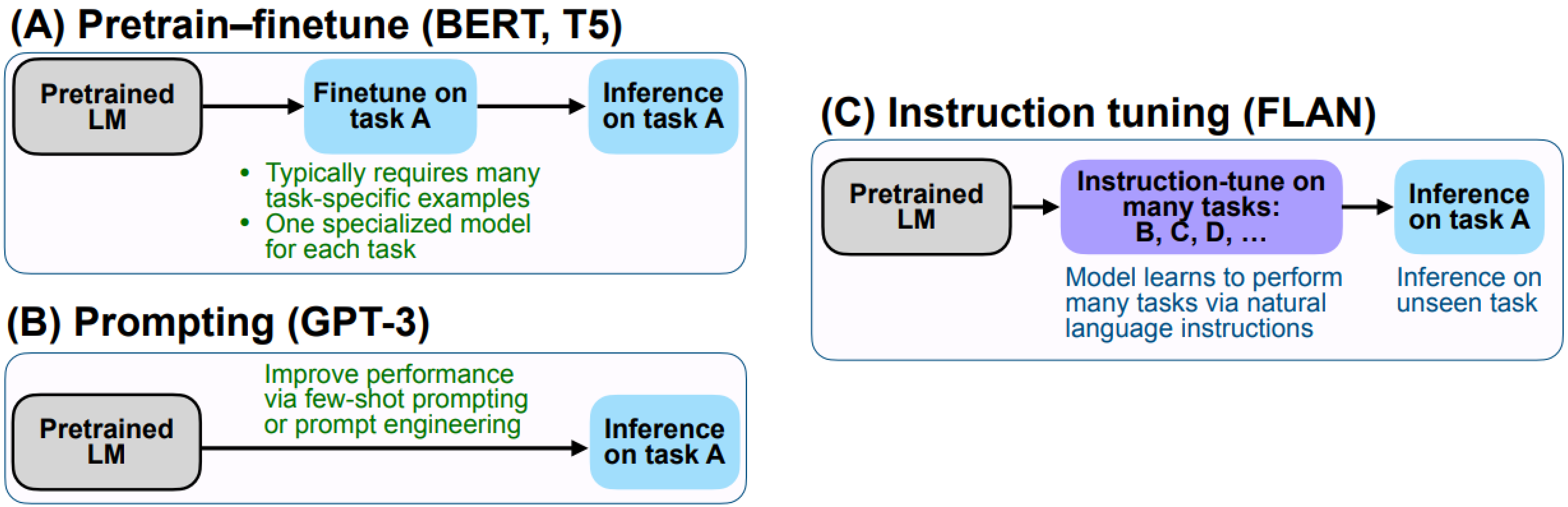
Google Research

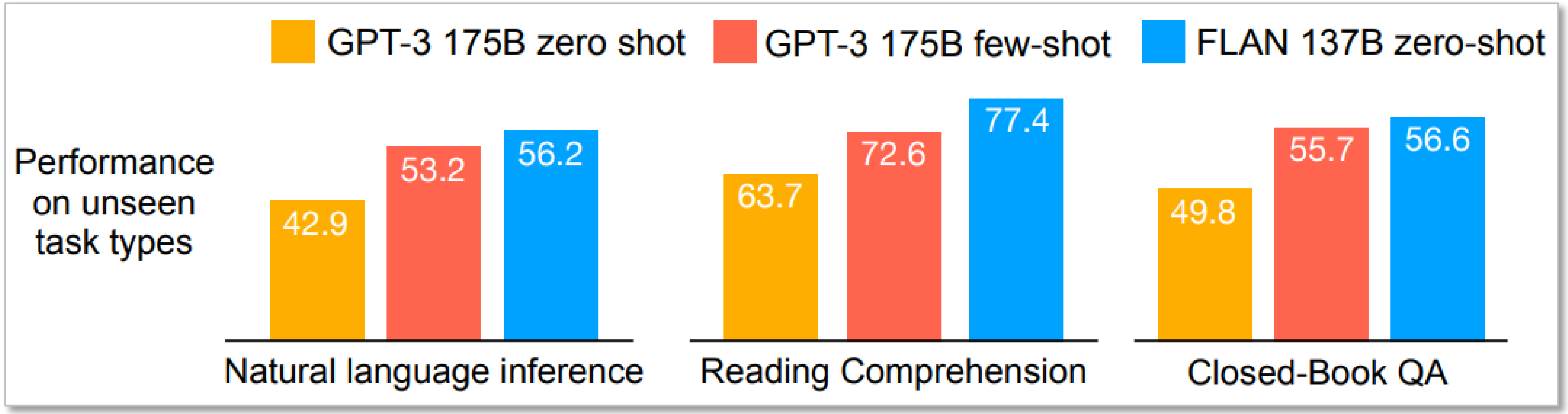
Abstract

This paper explores a simple method for improving the zero-shot learning abilities of language models. We show that *instruction tuning*—finetuning language models on a collection of datasets described via instructions—substantially improves zero-shot performance on unseen tasks.

We take a 137B parameter pretrained language model and instruction tune it on over 60 NLP datasets verbalized via natural language instruction templates. We evaluate this instruction-tuned model, which we call FLAN, on unseen task types. FLAN substantially improves the performance of its unmodified counterpart and surpasses zero-shot 175B GPT-3 on 20 of 25 datasets that we evaluate. FLAN even

FLAN


(A) Pretrain–finetune (BERT, T5)
Pretrained LM
Finetune on task A
Inference on task A
• Typically requires many task-specific examples
• One specialized model for each task
(B) Prompting (GPT-3)
Pretrained LM
Improve performance via few-shot prompting or prompt engineering
Inference on task A
(C) Instruction tuning (FLAN)
Pretrained LM
Instruction-tune on many tasks: B, C, D, …
Inference on task A
Model learns to perform many tasks via natural language instructions
Inference on unseen task


GPT-3 175B zero shot
GPT-3 175B few-shot
FLAN 137B zero-shot
Performance on unseen task types
42.9
53.2
56.2
63.7
72.6
77.4
49.8
55.7
56.6
Natural language inference
Reading Comprehension
Closed-Book QA

Instruction Tuning: A Competition of Data Collection/Annotation

(a) SUP-NATINST (this work)

(b) NATINST

(c) PROMPTSOURCE (T0 subset)

(d) FLAN

(e) INSTRUCTGPT

Sup-NatInst: 1,616 tasks

FLAN: 62 tasks

InstructGPT [Ouyang et al., NeurIPS 2022]

Training language models to follow instructions with human feedback

Long Ouyang* **Jeff Wu*** **Xu Jiang*** **Diogo Almeida*** **Carroll L. Wainwright***

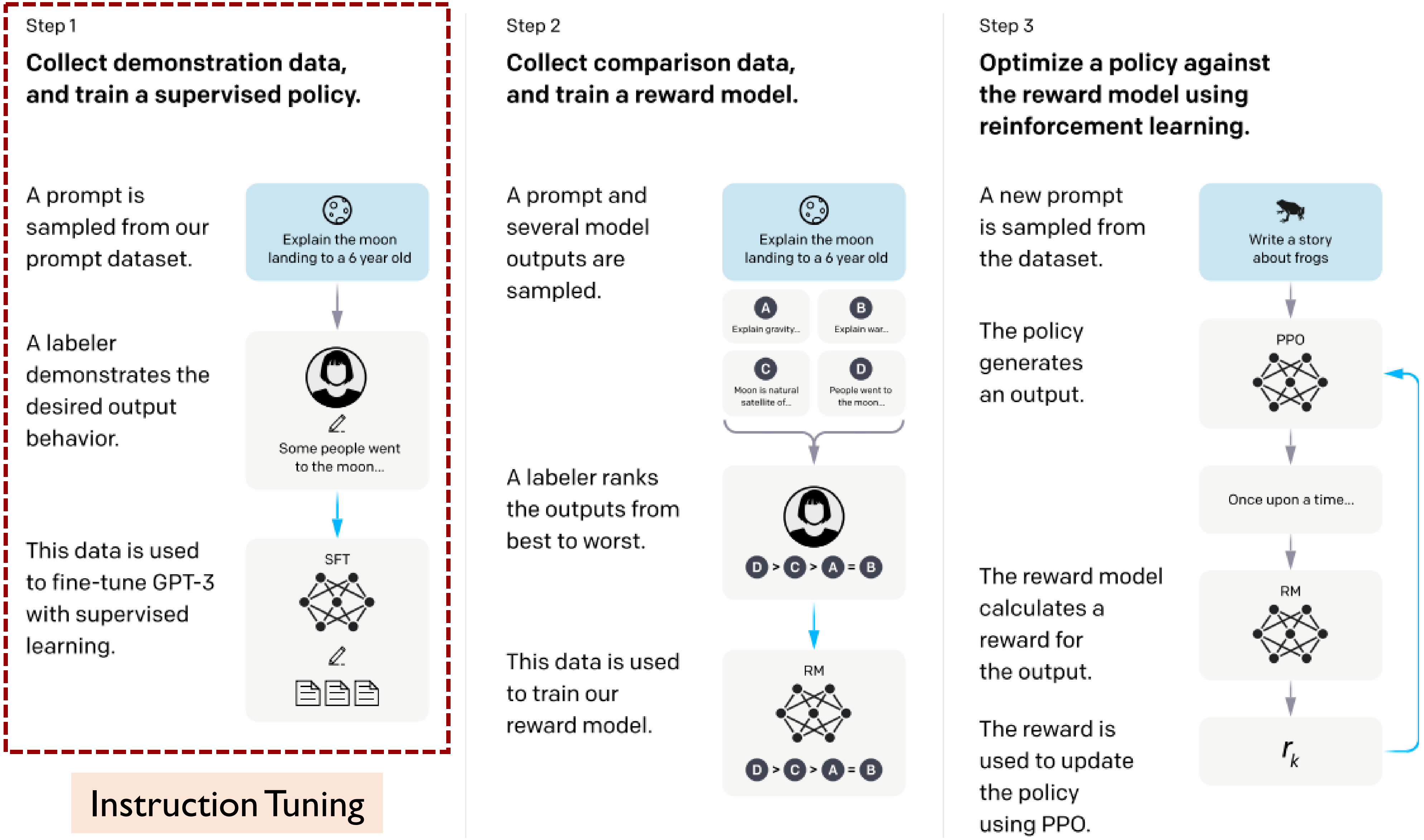
Pamela Mishkin* **Chong Zhang** **Sandhini Agarwal** **Katarina Slama** **Alex Ray**

John Schulman

Amanda A

OpenAI

InstructGPT

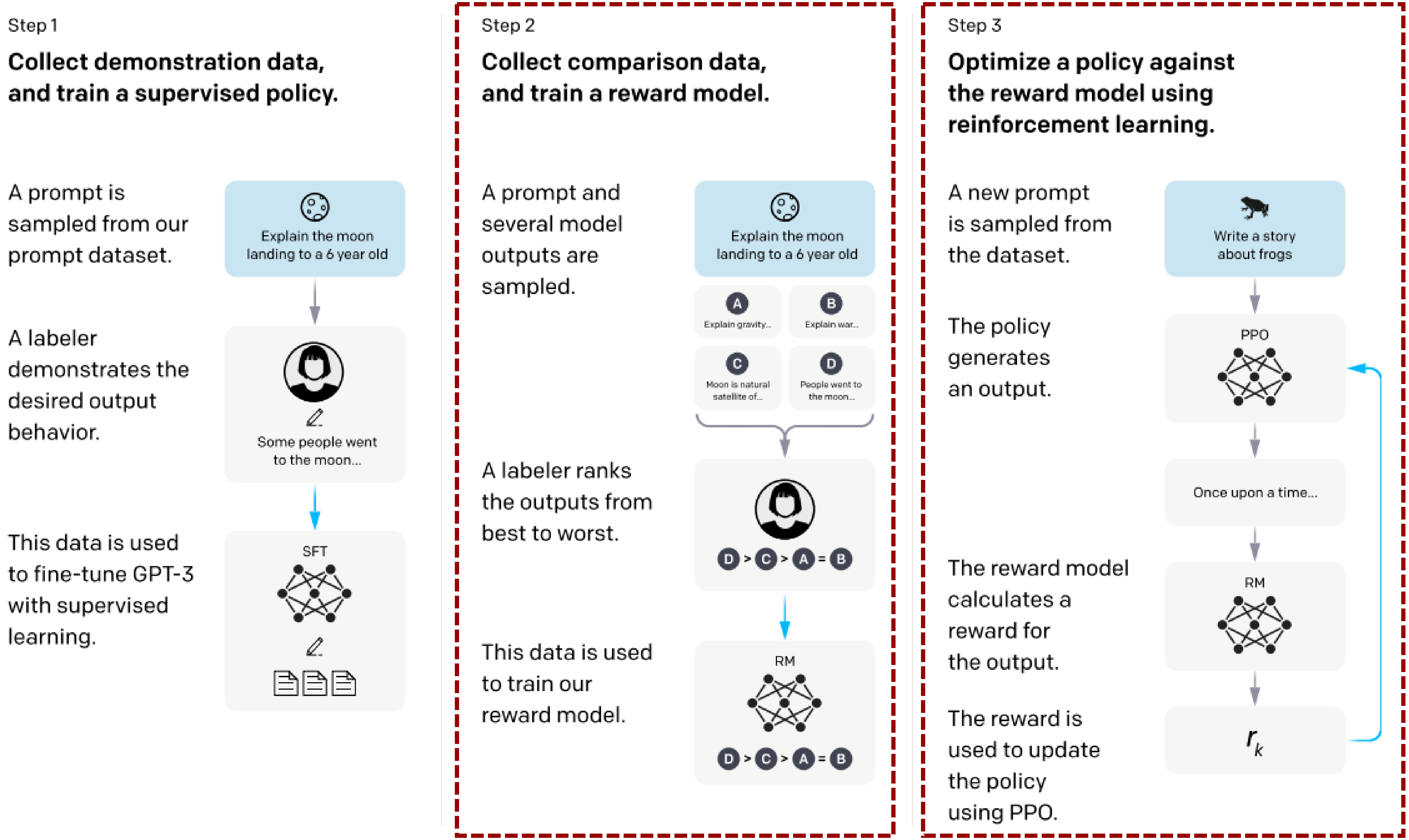


Instruction Tuning

InstructGPT

Train another model to imitate human preferences

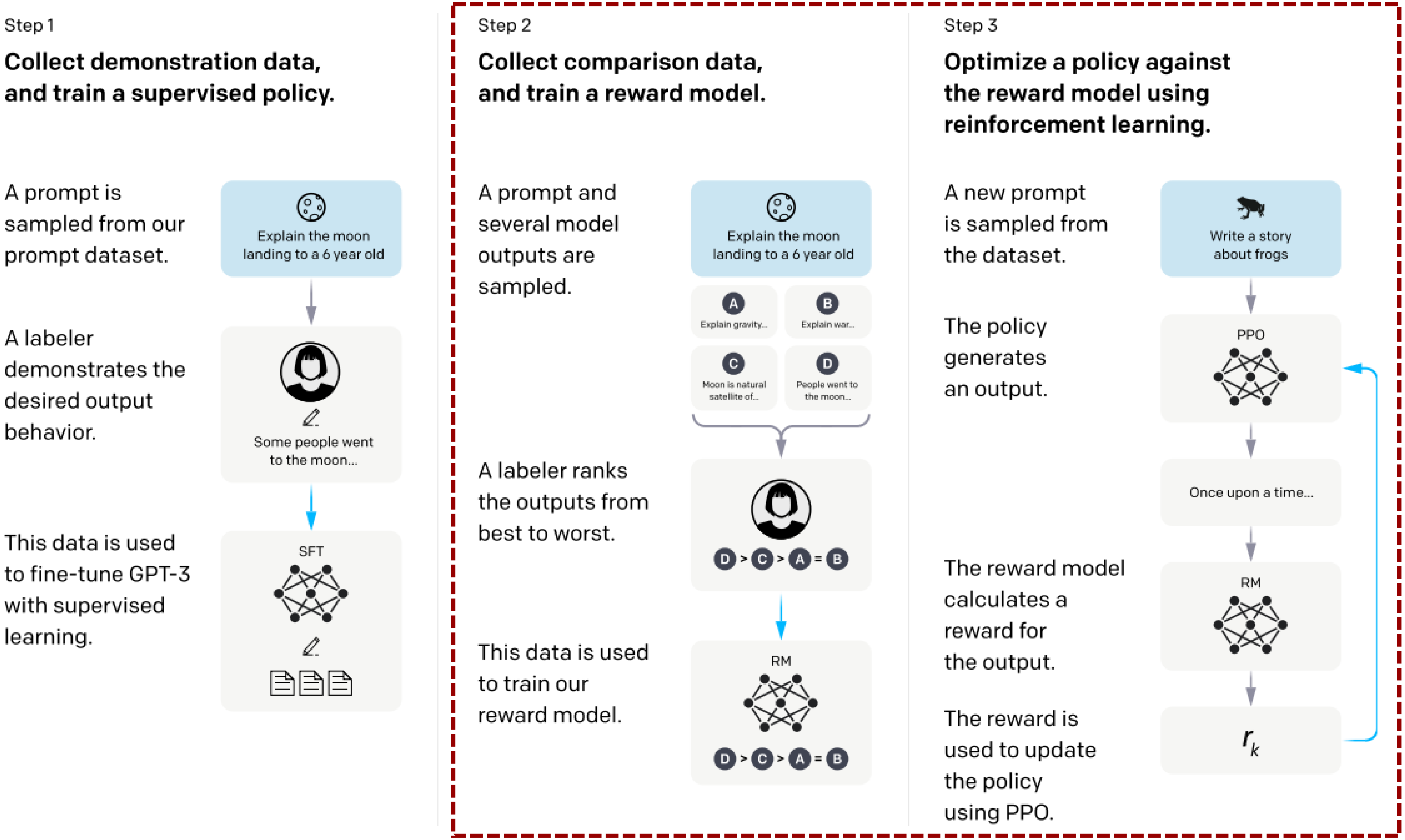
Train GPT to satisfy the "human" model



InstructGPT

Alignment

Reinforcement Learning with Human Feedback (RLHF)



What if an LLM does not have the knowledge to answer a question? (Retrieval-Augmented Generation and Tool Use)

What if an LLM does not have the knowledge to answer a question?

You

What is the most cited paper in WWW 2017? Please list its title and authors.

ChatGPT

As of my last update in January 2022, I don't have the real-time citation information, but one of the highly cited papers from WWW 2017 is:

Title: "HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning"

Authors: Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, Edward Y. Chang

Please verify the latest citation counts and updates from reliable sources or academic databases for the most current information.

HIN2Vec was published in CIKM 2017.

HIN2Vec was written by Tao-yang Fu, Wang-Chien Lee, and Zhen Lei.

ChatGPT 3.5, queried on January 23, 2024

What if an LLM does not have the knowledge to answer a question?

- Several reasons why this happens
  - Knowledge cutoffs: parameters are usually only updated to a particular time
  - Private data: data stored in private text or data repositories not suitable for training
  - Learning failures: even for data that the model was trained on, it might not be sufficient to get the right answer

Toolformer [Schick et al., NeurIPS 2023]

Toolformer: Language Models Can Teach Themselves to Use Tools

Timo Schick **Jane Dwivedi-Yu** **Roberto Dessì**[†] **Roberta Raileanu**
Maria Lomeli **Eric Hambro** **Luke Zettlemoyer** **Nicola Cancedda** **Thomas Scialom**

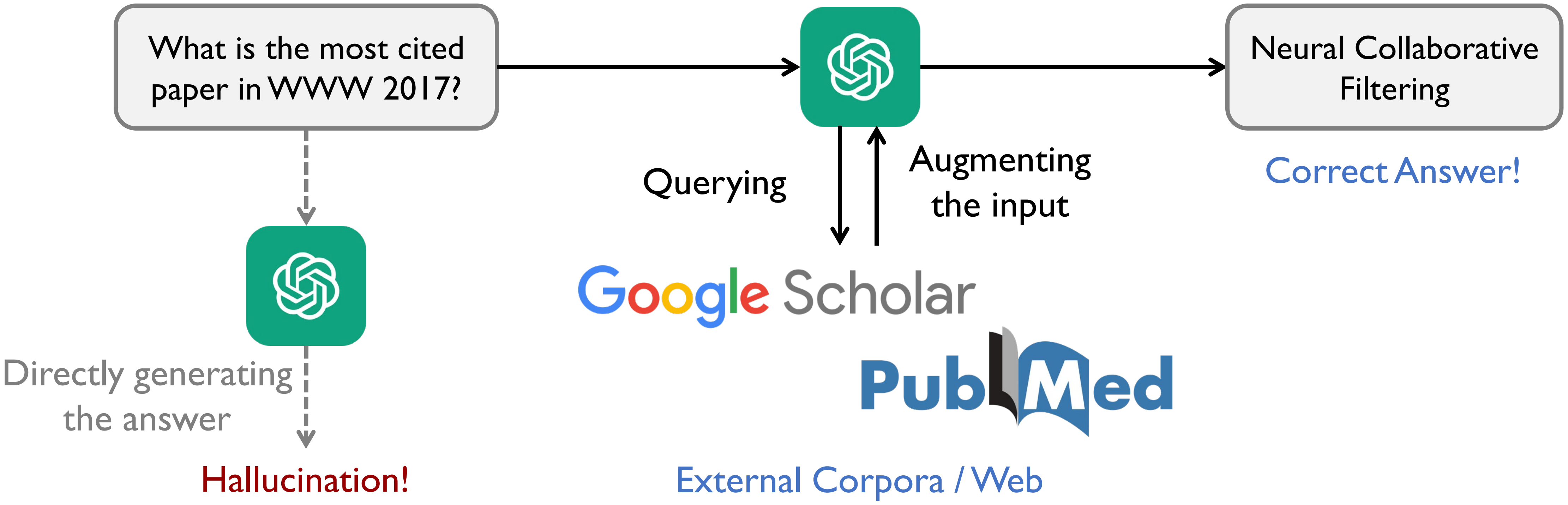
FAIR, Meta [†]Universitat Pompeu Fabra

Abstract

Language models (LMs) exhibit remarkable abilities to solve new tasks from just a few examples or textual instructions, especially at scale. They also, paradoxically, struggle with basic functionality, such as arithmetic or factual lookup, where much simpler and smaller specialized models excel. In this paper, we show that LMs can teach themselves to *use external tools* via simple APIs and achieve the best of both worlds. We introduce *Toolformer*, a model trained to decide which APIs to call, when to call them, what arguments to pass, and how to best incorporate the results into future token prediction. This is done in a self-supervised way, requiring

Retrieval-Augmented Generation

- *"Give a man a fish, and you feed him for a day; teach a man to fish, and you feed him for a lifetime."*
- *"Teach an LLM to know one more fact, and you help it answer one question; teach an LLM to explore the external corpora/Web, and you help it answer many questions."*



What if retrieval is not enough?

What is the most cited paper in WWW 2017? Please list its title and authors.

Searched 5 sites

The most cited paper from WWW 2017 is "Attention is All You Need," which introduced the Transformer model, a groundbreaking architecture in natural language processing that uses attention mechanisms without relying on recurrent or convolutional layers. The authors of the paper are Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin ARSIV NEURIPS PROCEEDINGS.

Transformer was published in NeurIPS 2017.

ChatGPT 4o, queried on October 12, 2024

What if retrieval is not enough?

- Retrieval is just one "tool" that can be used by LLMs.
- There are many other "tools".

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

- Tool use: switching between the text-generation mode and the tool-execution mode

How to induce tool use?

- Training
- Inference-time prompting

Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.
Output: Joe Biden was born in **[QA("Where was Joe Biden born?")]** Scranton, **[QA("In which state is Scranton?")]** Pennsylvania.

Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.
Output: Coca-Cola, or **[QA("What other name is Coca-Cola known by?")]** Coke, is a carbonated soft drink manufactured by **[QA("Who manufactures Coca-Cola?")]** the Coca-Cola Company.

Input: x
Output:

What if an LLM has access to the knowledge necessary to answer a question, but arriving at the solution requires a complex reasoning process?

LLM Reasoning

- The correct answer (finally…):

What is the most cited paper in WWW 2017? Please list its title and authors.

Reasoning
Started thinking...

The most cited paper from the **WWW 2017** conference is:

Title: "*Neural Collaborative Filtering*"

Authors: Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua

DeepSeek-R1 671B, queried on May 8, 2025

Scratchpad of DeepSeek-R1

Reasoning
Started thinking...

Okay, so I need to find out the most cited paper from the WWW 2017 conference. Let me start by recalling that WWW stands for the International World Wide Web Conference, which is a major venue for research related to the web. The 2017 edition would have been the 26th conference.

First, I should figure out how to determine the most cited paper from that specific year. Citation counts can be tracked through databases like Google Scholar, Semantic Scholar, or the ACM Digital Library since ACM often sponsors these conferences. However, citation counts can vary over time, so the most cited paper might change as newer papers accumulate citations. But since the user is asking about the current most cited, I need up-to-date information.

I'll start by checking the proceedings of WWW 2017. The conference usually publishes its proceedings, and sometimes the awards or notable papers are mentioned. If I can find the list of accepted papers, I can then look up each of them on Google Scholar to check their citation counts. But that's time-consuming. Alternatively, maybe there's a known paper from that year that's highly cited. For example, in recent years, transformer-based models like BERT or attention mechanisms have been highly cited, but 2017 might be a bit early for those.

Wait, 2017 is the year when the "Attention Is All You Need" paper was published, but that was at NIPS, not WWW. So maybe that's not relevant here. Let me think. In the realm of web conferences, topics like graph neural networks, recommendation systems, network analysis, and so on are common.

Another approach is to search for "most cited papers from WWW 2017" on Google or academic search engines. Let me try that. Searching for that exact phrase, I find some resources. For instance, there's a page on the WWW 2017 conference website that lists accepted papers, but I need to check if they have any awards or most cited mentions. Alternatively, Semantic Scholar might have a list sorted by citations.

Let me check Semantic Scholar. Going to their website, I can search for papers from the WWW 2017 conference. Applying the filters for the year 2017 and the venue WWW, then sorting by citations. Let's see. The top result there might be the one. Alternatively, Google Scholar can be used by searching for the conference name and year, then sorting by citations.

Upon doing a quick search, I find that the paper "Inductive Representation Learning on Large Graphs" by William L. Hamilton, Rex Ying, and Jure Leskovec is highly cited. Wait, but that was published at NIPS 2017, right? Wait, no, maybe I'm confused. Let me check the details. The Hamilton et al. paper on GraphSAGE was indeed presented at NIPS 2017. So maybe that's not the one.

Alternatively, another paper from WWW 2017 that's highly cited could be related to graph convolutional networks or something similar. Let me check the proceedings. For example, "Graph Convolutional Matrix Completion" by Rianne van den Berg, Thomas N. Kipf, and Max Welling was published in 2017, but I think that's at a different conference.

Another possibility is "Neural Collaborative Filtering" by Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Wait, but that paper was published in WWW 2017? Let me confirm. According to Google Scholar, that paper was indeed published in the Proceedings of the 26th International Conference on World Wide Web in 2017. It's possible that this paper is highly cited as it introduces neural networks into collaborative filtering, a popular approach in recommendations.

Checking the citation counts: As of now, "Neural Collaborative Filtering" has over 5,000 citations on Google Scholar, which is very high. Another paper, "DeepFM: A Factorization-Machine based Neural Network for CTR Prediction" has over 2,000 citations. There might be others. Wait, but I need to confirm which one has the highest.

Alternatively, "Inductive Representation Learning on Large Graphs" (GraphSAGE) by Hamilton et al. has over 6,000 citations, but that's from NIPS 2017, not WWW. So perhaps the most cited from WWW 2017 is "Neural Collaborative Filtering". Let me cross-verify.

Looking at the WWW 2017 accepted papers list, "Neural Collaborative Filtering" is indeed there. The authors are Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. That paper is foundational in combining neural networks with collaborative filtering, which explains its high citation count.

Alternatively, another paper from WWW 2017 that's highly cited is "StarSpace: Embed All The Things!" by Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. This paper also has a substantial number of citations, but I think it's less than "Neural Collaborative Filtering".

To be sure, checking Google Scholar for "Neural Collaborative Filtering" shows over 5,000 citations as of 2023. "DeepFM" has around 2,000. Another paper, "Metapath2vec: Scalable Representation Learning for Heterogeneous Networks" by Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami, which was also presented at WWW 2017, has over 1,500 citations.

Therefore, it's likely that "Neural Collaborative Filtering" by He et al. is the most cited paper from WWW 2017. However, I should also check if there's any other paper with higher citations. For example, "Attention Is All You Need" is from 2017 but not WWW. "Dynamic Routing Between Capsules" by Hinton et al. is also from 2017 but a different conference.

Another possible candidate is "Semi-Supervised Classification with Graph Convolutional Networks" by Kipf and Welling, but that was published at ICLR 2017, so not WWW.

Therefore, after checking the citation counts and cross-referencing with the conference proceedings, "Neural Collaborative Filtering" by Xiangnan He and colleagues seems to be the most cited paper from WWW 2017. However, I should verify once more. Let me check another source.

On Semantic Scholar, searching for WWW 2017 papers sorted by citations: the top result is "Neural Collaborative Filtering" with 5,442 citations as of October 2023. The next is "Metapath2vec" with 2,035 citations. Then "DeepFM" with 1,821. So yes, "Neural Collaborative Filtering" is the most cited.

Thus, the answer should be that paper.

Scratchpad of DeepSeek-R1

Another approach is to search for "most cited papers from WWW 2017" on Google or academic search engines. Let me try that. Searching for that exact phrase, I find some resources. For instance, there's a page on the WWW 2017 conference website that lists accepted papers, but I need to check if they have any awards or most cited mentions. Alternatively, Semantic Scholar might have a list sorted by citations.

Let me check Semantic Scholar. Going to their website, I can search for papers from the WWW 2017 conference. Applying the filters for the year 2017 and the venue WWW, then sorting by citations. Let's see. The top result there might be the one. Alternatively, Google Scholar can be used by searching for the conference name and year, then sorting by citations.

Checking the citation counts: As of now, "Neural Collaborative Filtering" has over 5,000 citations on Google Scholar, which is very high. Another paper, "DeepFM: A Factorization-Machine based Neural Network for CTR Prediction" has over 2,000 citations. There might be others. Wait, but I need to confirm which one has the highest.

Alternatively, "Inductive Representation Learning on Large Graphs" (GraphSAGE) by Hamilton et al. has over 6,000 citations, but that's from NIPS 2017, not WWW. So perhaps the most cited from WWW 2017 is "Neural Collaborative Filtering". Let me cross-verify.

DeepSeek-R1 [Guo et al., Nature 2025]

Article | Open access | Published: 17 September 2025

DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, ... Zhen Zhang + Show authors

Nature **645**, 633–638 (2025) | Cite this article

Abstract

General reasoning represents a long-standing and formidable challenge in artificial intelligence (AI). Recent breakthroughs, exemplified by large language models (LLMs)[1,2] and

How to force LLMs to reason?

Supervised Fine-Tuning (SFT):
Imitating surface-level patterns from training data

→

Reinforcement Learning (RL):
Maximize long-term reward (e.g., correctness, format)

- Supervised Fine-Tuning (SFT): Given a question, maximize the likelihood that the LLM generates the reference answer.
    - If the reference answer is 0.5, then generating 1/2 is wrong.
    - If the LLM is asked to write code for a specific function, only an exact match with the reference answer is considered fully correct. A program that differs from the reference by just one token (even if it does not run at all) is regarded as better than an alternative implementation that correctly achieves the same functionality.

How to force LLMs to reason?

Supervised Fine-Tuning (SFT):
Imitating surface-level patterns from training data

→

Reinforcement Learning (RL):
Maximize long-term reward (e.g., correctness, format)

- Reinforcement Learning with Verifiable Rewards (RLVR): When the LLM does something correct according to our rubrics, it gets certain reward. The model is trained to maximize the reward.
  - Format Reward:
    - Has <think> and </think> tokens in its output; puts its intermediate reasoning steps between these two tokens.
    - Has <answer> and </answer> tokens in its output; puts its final answer between these tokens
    - …

How to force LLMs to reason?

Supervised Fine-Tuning (SFT): Imitating surface-level patterns from training data

→

Reinforcement Learning (RL): Maximize long-term reward (e.g., correctness, format)

- Reinforcement Learning with Verifiable Rewards (RLVR): When the LLM does something correct according to our rubrics, it gets certain reward. The model is trained to maximize the reward.
  - Correctness Reward:
    - The answer (i.e., tokens between <answer> and </answer>) should be "equivalent" to the reference answer (according to a verifier).
  - If the model does not generate the correct answer, it will NOT know the answer after this iteration; it will only know that the reward is low.
    - This question can be used repeatedly until the model generates a good answer.

Curious about the details of RLVR or R1-style models?

- Next lecture!
- Guest lecture (online! https://tamu.zoom.us/j/6411788612)

Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning

Bowen Jin[1], **Hansi Zeng**[2], **Zhenrui Yue**[1], **Jinsung Yoon**[3], **Sercan Ö. Arık**[3], **Dong Wang**[1], **Hamed Zamani**[2], **Jiawei Han**[1]
[1] Department of Computer Science, University of Illinois at Urbana-Champaign
[2] Center for Intelligent Information Retrieval, University of Massachusetts Amherst
[3] Google Cloud AI Research
{bowenj4,zhenrui3,dwang24,hanj}@illinois.edu, {hzeng, zamani}@cs.umass.edu
{jinsungyoon,soarik}@google.com

Thank You!

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html