



RAISE AI Bridge Summit @ Texas A&M

Large Language Models (Part II)

Yu Zhang

Assistant Professor, Department of Computer Science & Engineering

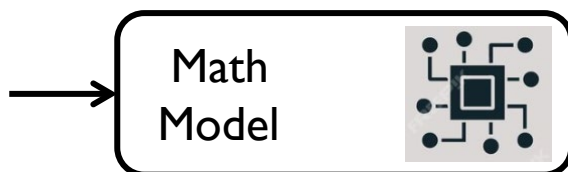
yuzhang@tamu.edu

January 6, 2026

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

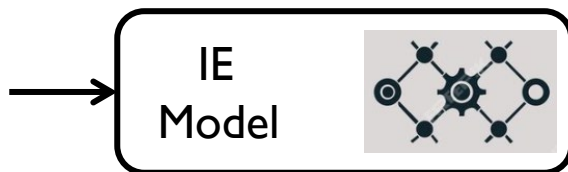
Math Word Problem: (John buys 20 cards and 1/4 are uncommon. How many uncommon cards did he get?, 5)



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



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

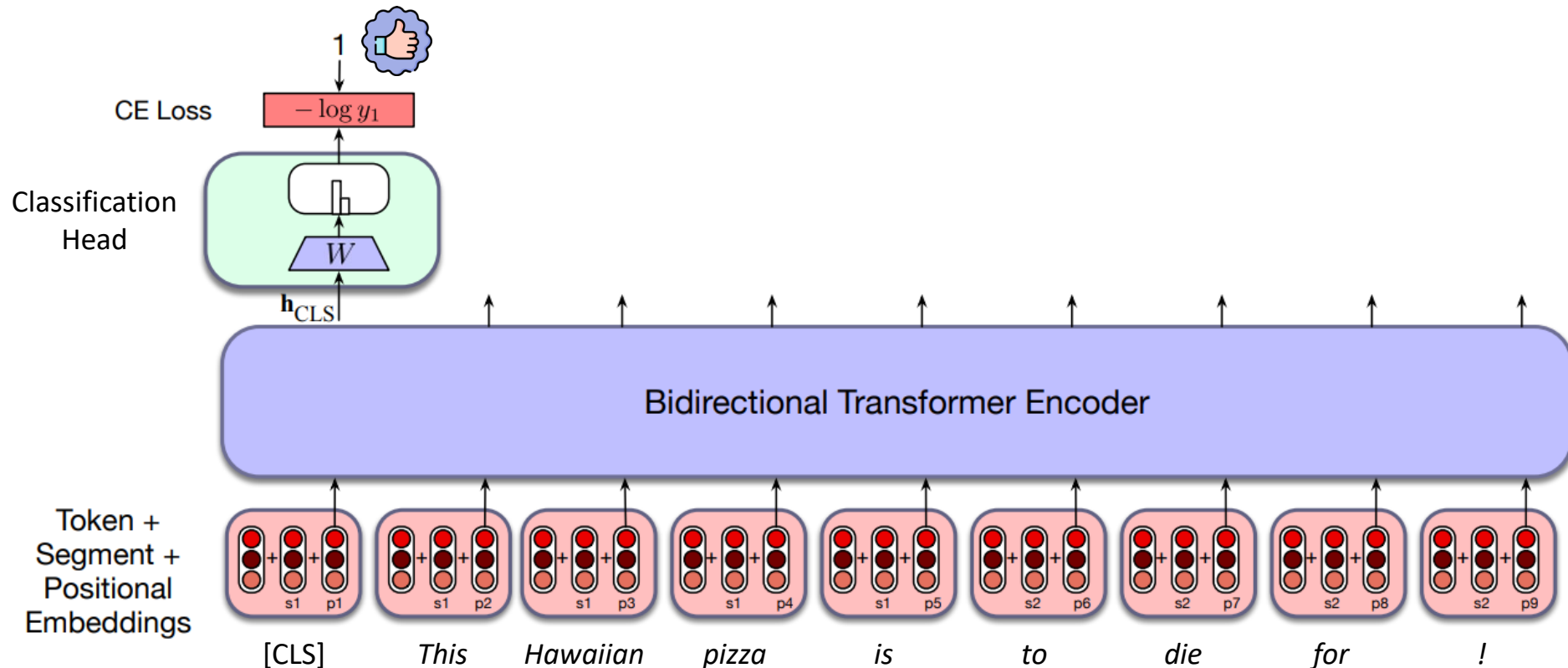


Question Answering: (Who formulated the zeroth law of thermodynamics?, Ralph H. Fowler)



Recap: BERT [Devlin et al., 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.



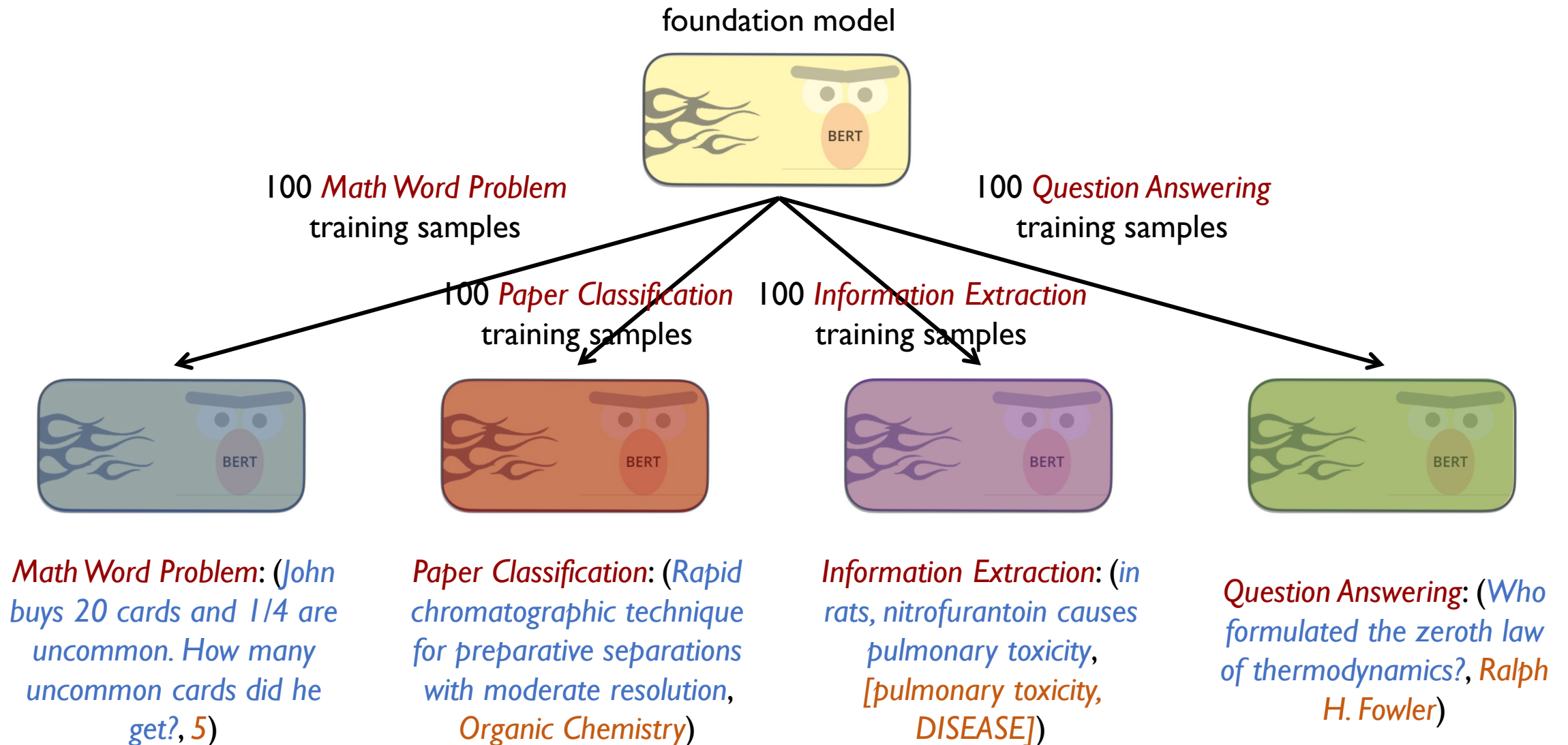
How to fine-tune BERT for different NLP tasks?

- *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

<i>in</i>	<i>rats</i>	<i>,</i>	<i>nitrofurantoin</i>	<i>causes</i>	<i>pulmonary</i>	<i>toxicity</i>
<i>None</i>	<i>None</i>	<i>None</i>	<i>B-Chemical</i>	<i>None</i>	<i>B-Disease</i>	<i>I-Disease</i>

- 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



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

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

0 Paper Classification training samples

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

0 Information Extraction training samples

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

0 Question Answering training samples



One Model for All Tasks!

5

Organic Chemistry

pulmonary toxicity

Ralph H. Fowler

Let's take a step back first

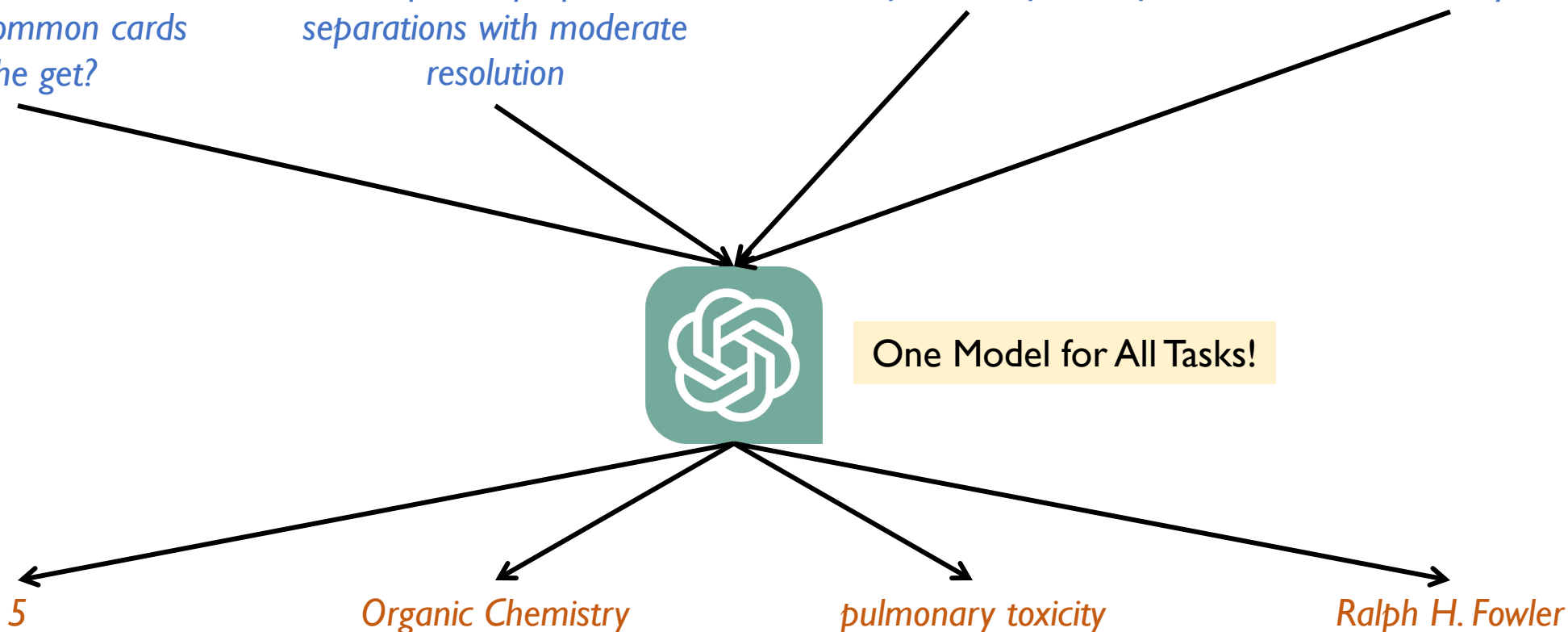
The **task instruction** is optional if you have 5 training samples.

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?



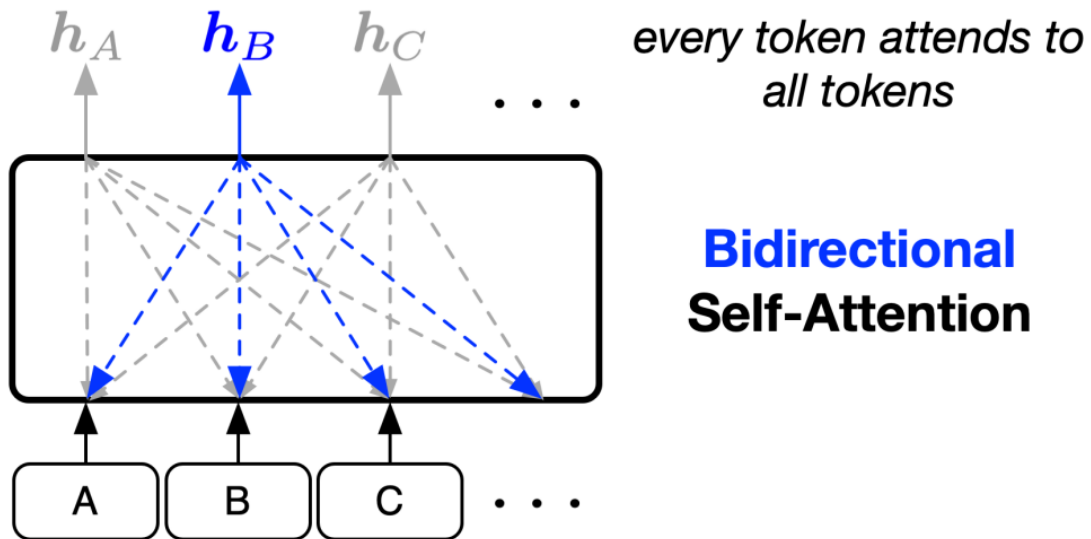
GPT-3 [Brown et al., 2020]

Language Models are Few-Shot Learners

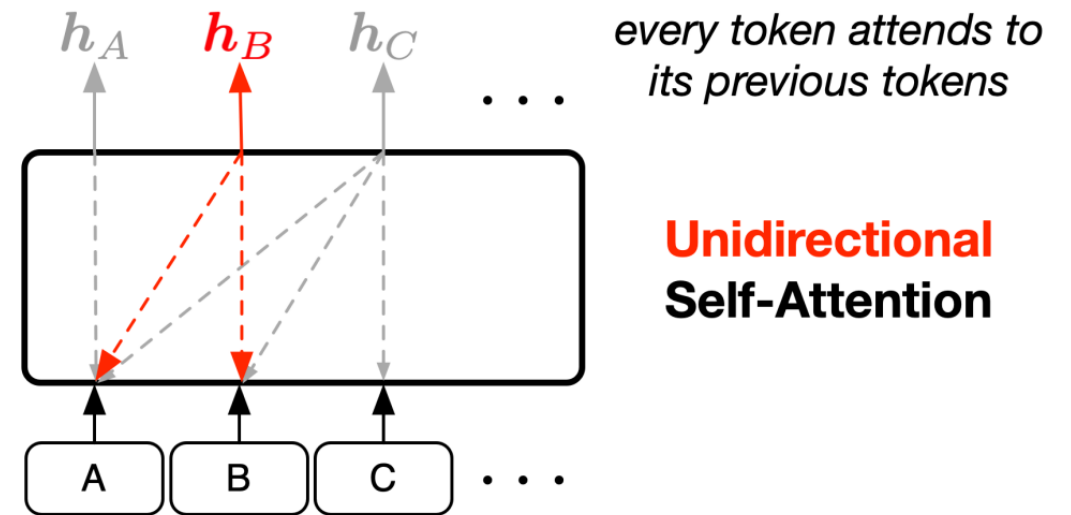
Tom B. Brown*	Benjamin Mann*	Nick Ryder*	Melanie Subbiah*	
Jared Kaplan[†]	Prafulla Dhariwal	Arvind Neelakantan	Pranav Shyam	Girish Sastry
Amanda Askell	Sandhini Agarwal	Ariel Herbert-Voss	Gretchen Krueger	Tom Henighan
Rewon Child	Aditya Ramesh	Daniel M. Ziegler	Jeffrey Wu	Clemens Winter
Christopher Hesse	Mark Chen	Eric Sigler	Mateusz Litwin	Scott Gray
Benjamin Chess		Jack Clark	Christopher Berner	
Sam McCandlish	Alec Radford	Ilya Sutskever	Dario Amodei	

OpenAI

Recap: Two Types of Transformer Architecture



$q1 \cdot k1$	$q1 \cdot k2$	$q1 \cdot k3$	$q1 \cdot k4$
$q2 \cdot k1$	$q2 \cdot k2$	$q2 \cdot k3$	$q2 \cdot k4$
$q3 \cdot k1$	$q3 \cdot k2$	$q3 \cdot k3$	$q3 \cdot k4$
$q4 \cdot k1$	$q4 \cdot k2$	$q4 \cdot k3$	$q4 \cdot k4$



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

What if we pre-train a Transformer decoder?

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

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

- 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?

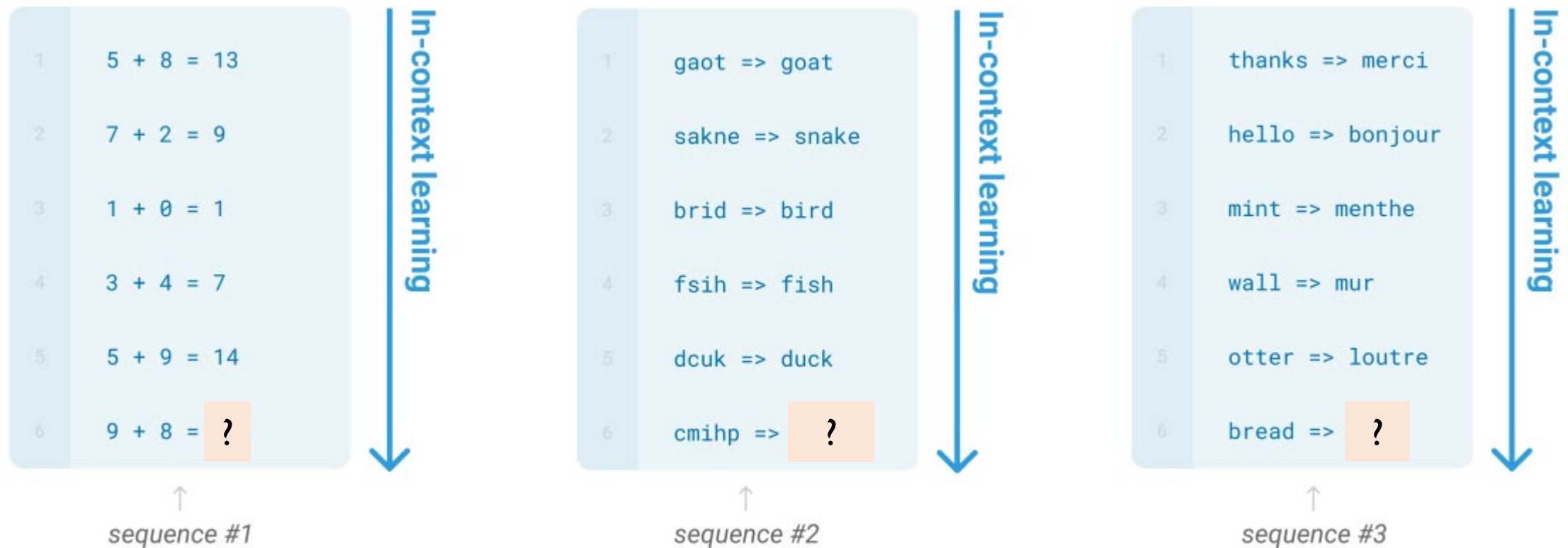
- With only one task – next token prediction – on web-scale corpora

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

- 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

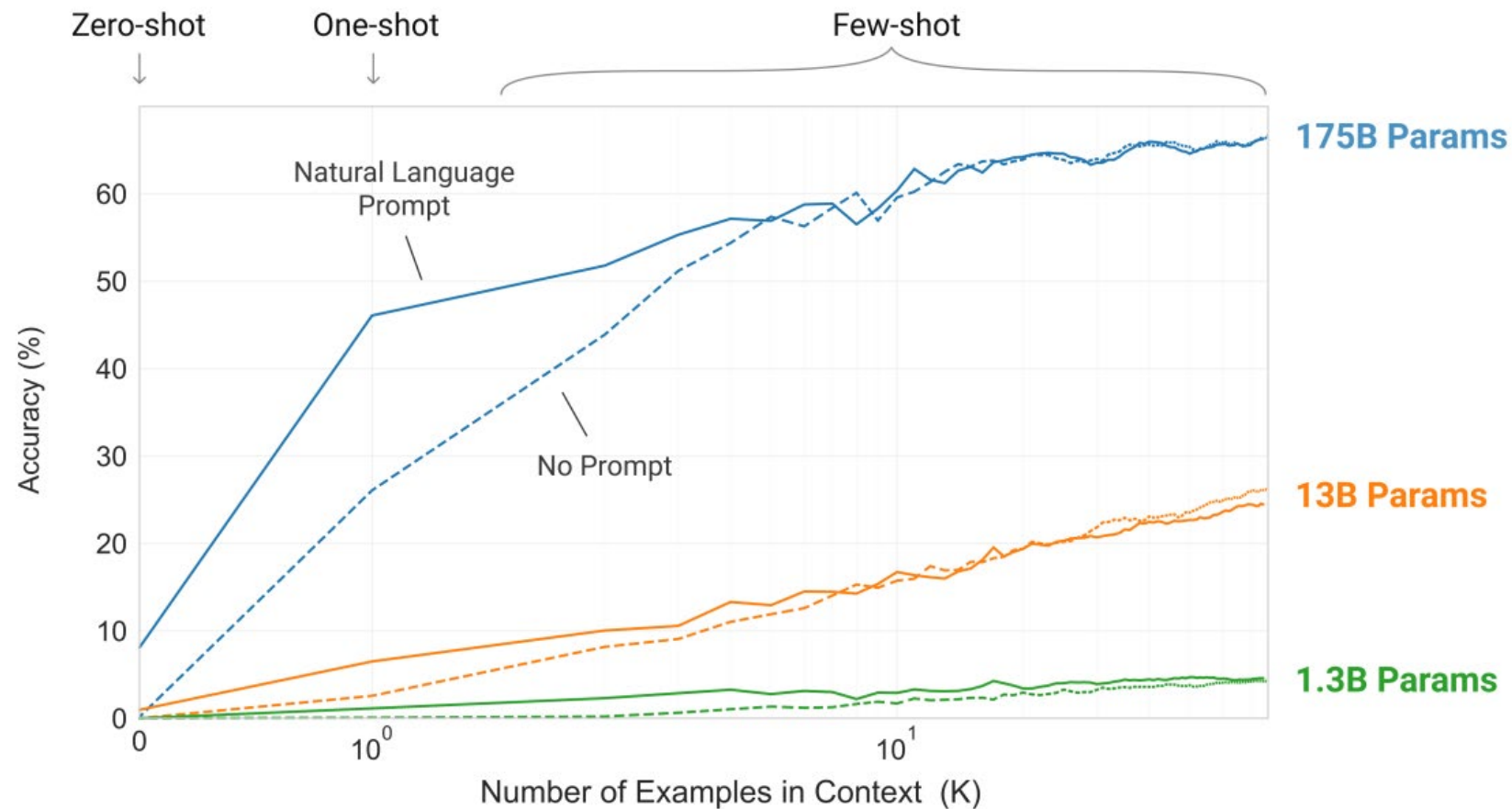
GPT-3 aims to perform each task with just a few examples

- 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!



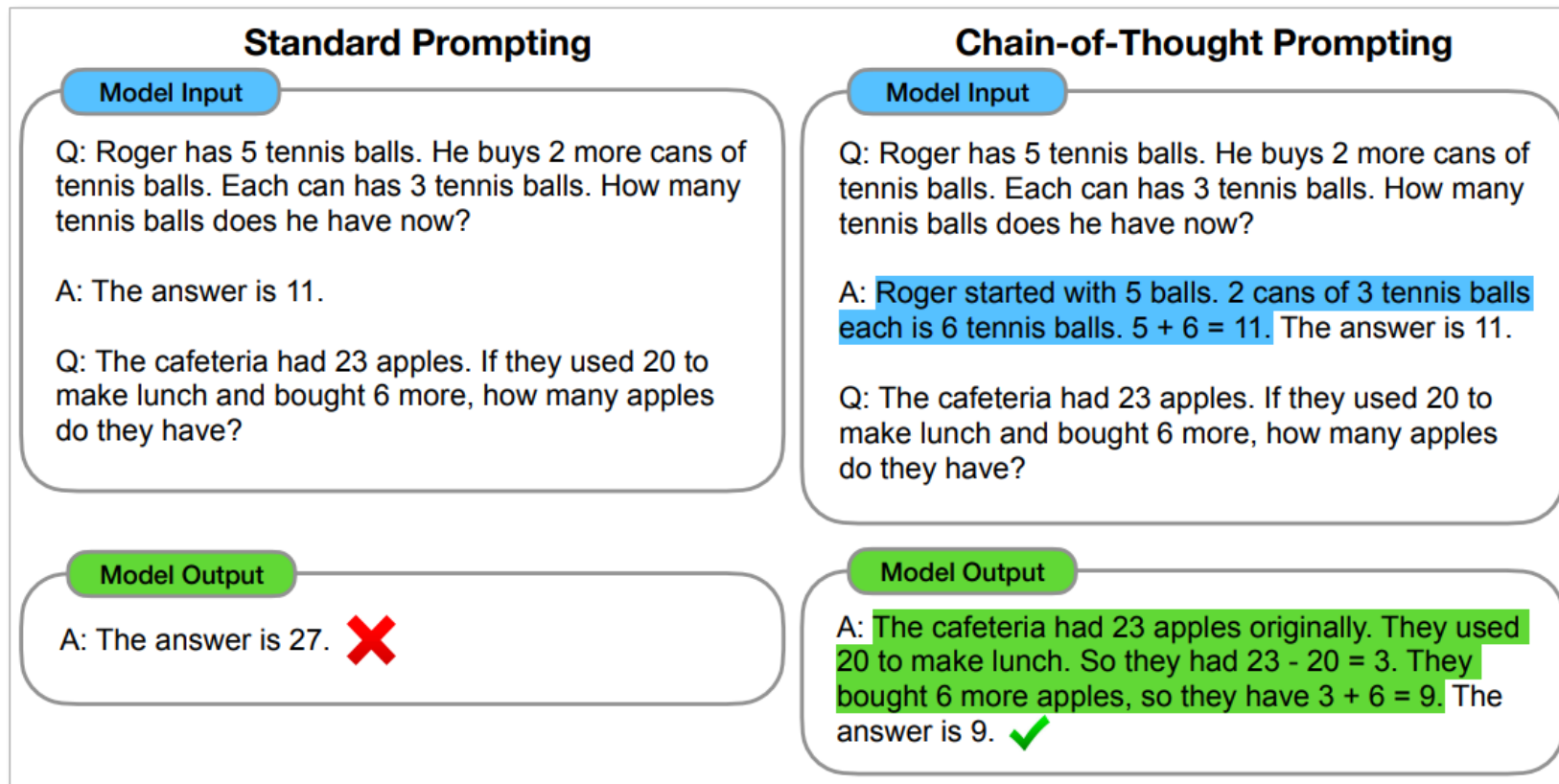
GPT-3

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

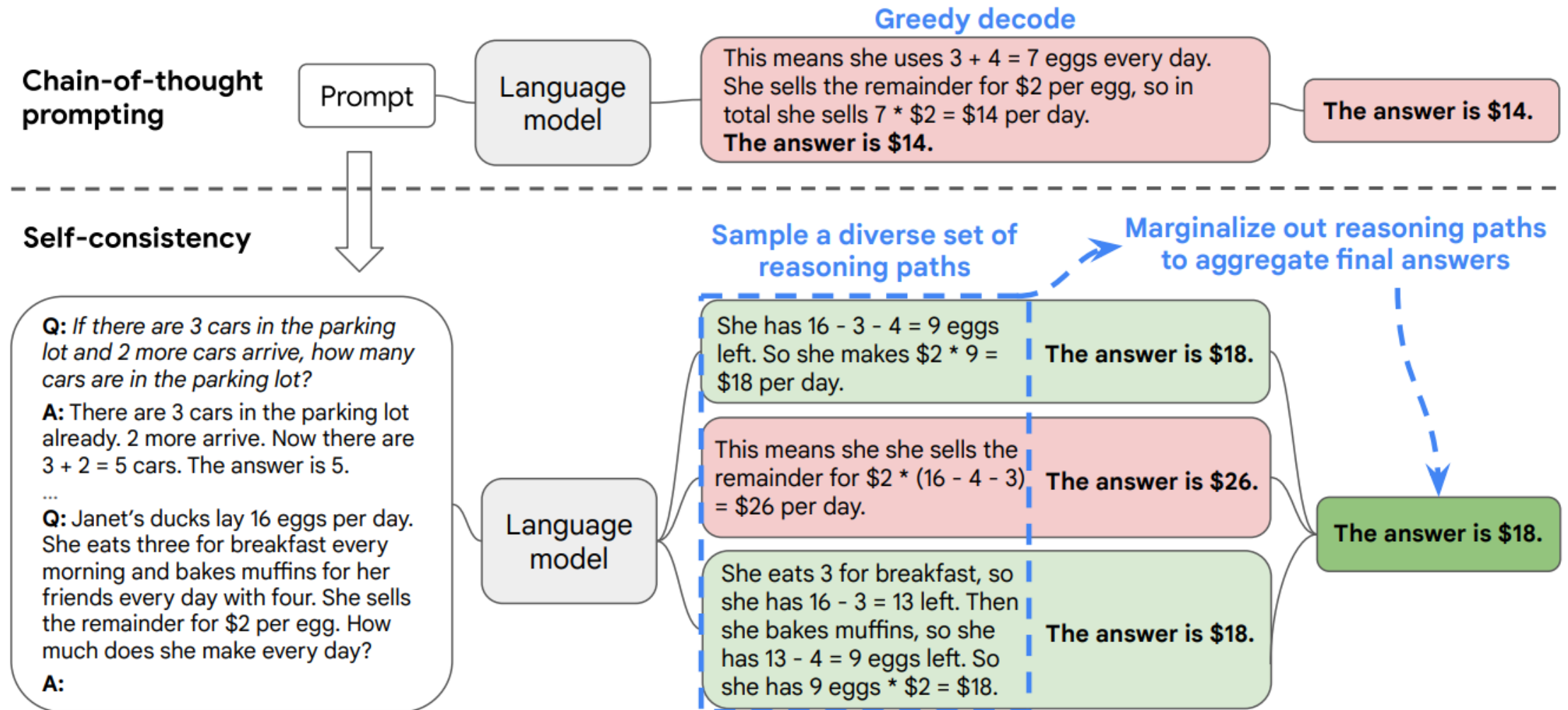
Can we make our demo examples more insightful?
(Prompt Engineering)

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

- 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., 2023]



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

Our Final Goal

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

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

0 Paper Classification training samples

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

0 Information Extraction training samples

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

0 Question Answering training samples



One Model for All Tasks!

5

Organic Chemistry

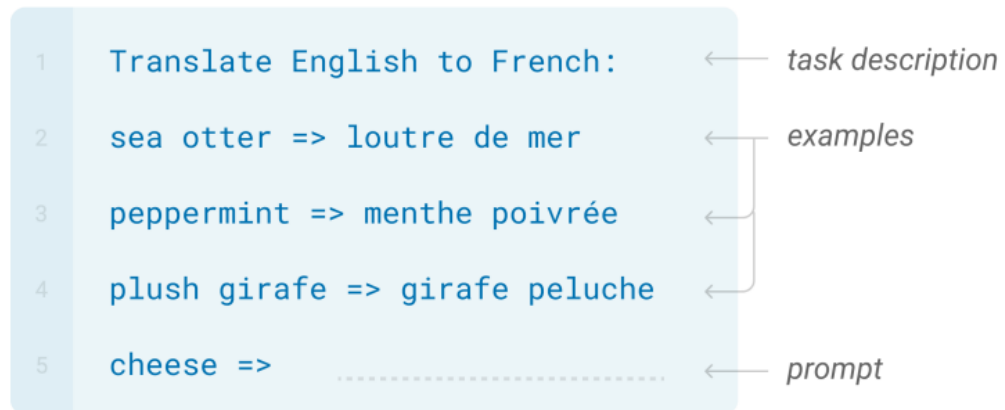
pulmonary toxicity

Ralph H. Fowler

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.

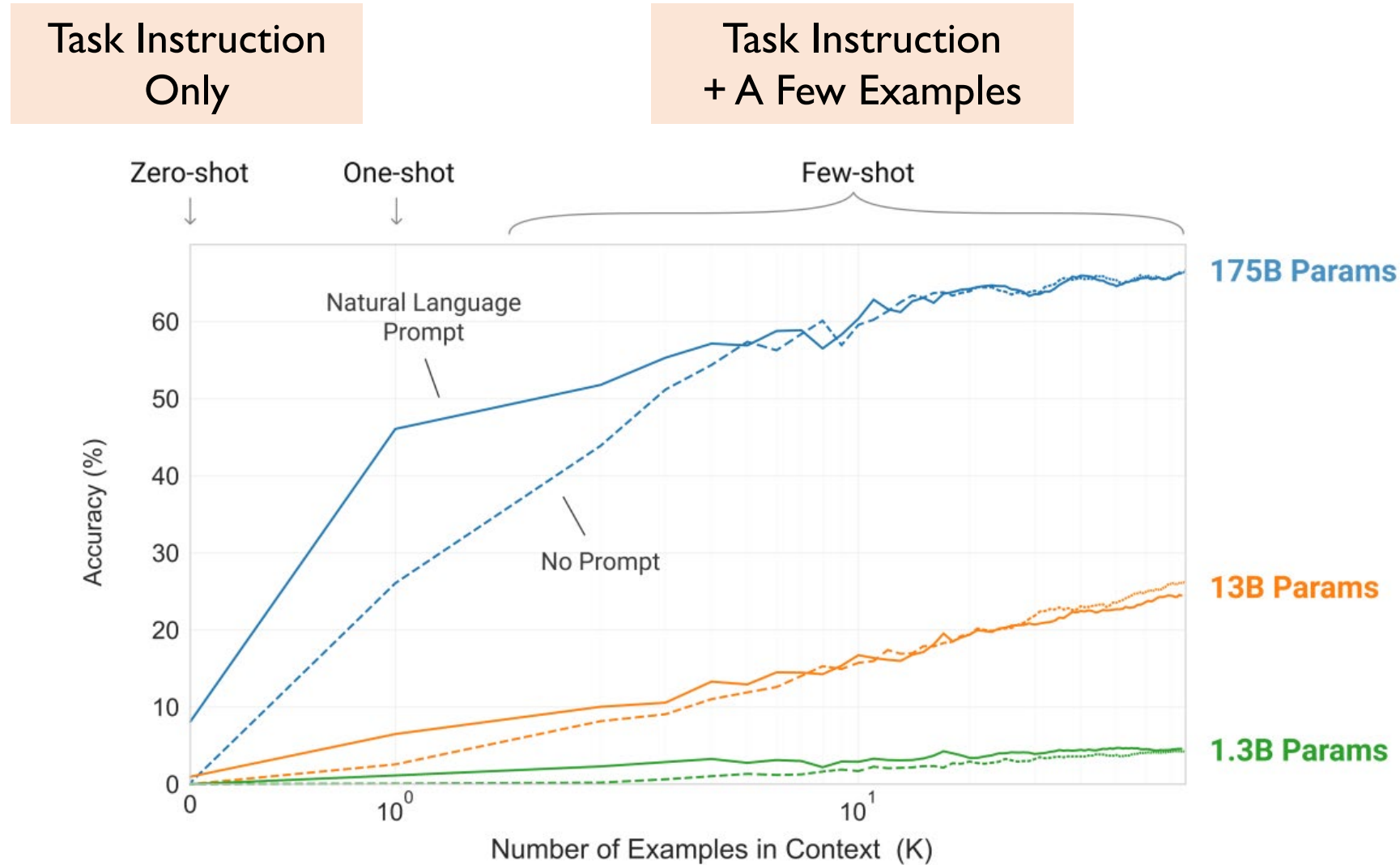


Zero-shot

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



Is GPT-3 a zero-shot learner?



Why is the zero-shot setting hard?

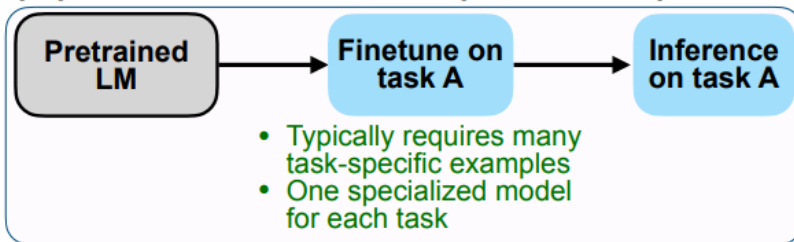
- GPT-3 is not good at following an instruction to perform a new task.
 - Because it is never asked to do so during pre-training.
- How to solve this problem?
 - Tune the model to follow task instructions!

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

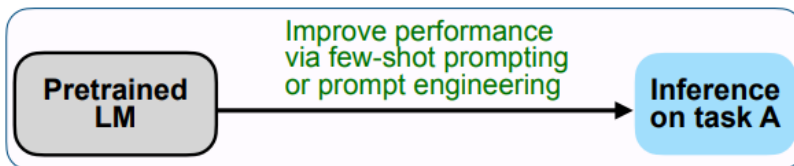
- How can we ensure that the model can handle unseen task instructions when it is actually used?
 - Let it see a sufficiently diverse set of task instructions during training so that it can generalize well
 - What really matters is the number of different tasks, not how many samples each task has
 - 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., 2022]

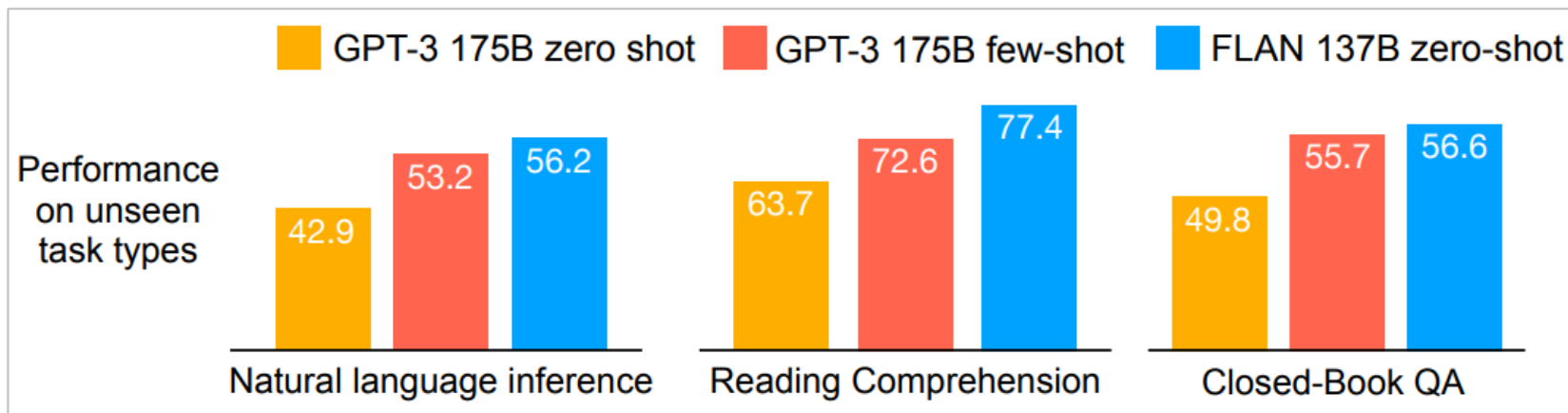
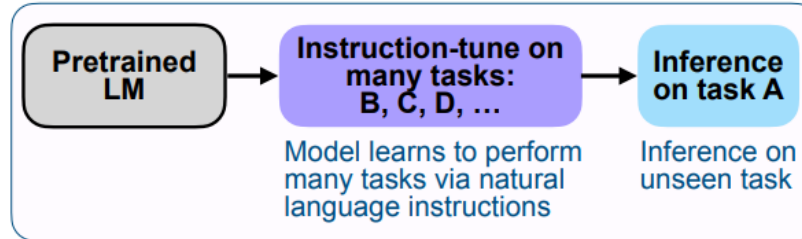
(A) Pretrain–finetune (BERT, T5)



(B) Prompting (GPT-3)



(C) Instruction tuning (FLAN)

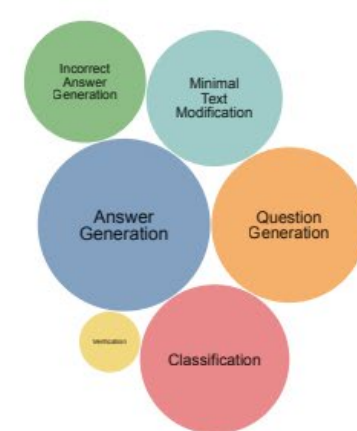


Instruction Tuning: A Competition of Data Collection/Annotation



(a) SUP-NATINST

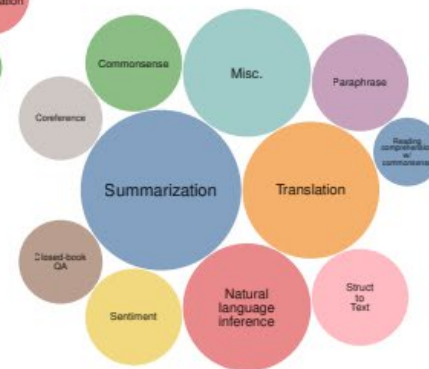
Sup-NatInst: 1,616 tasks



(b) NATINST

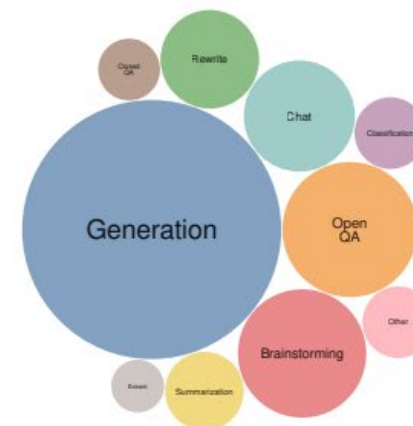


(c) PROMPTSOURCE (T0 subset)



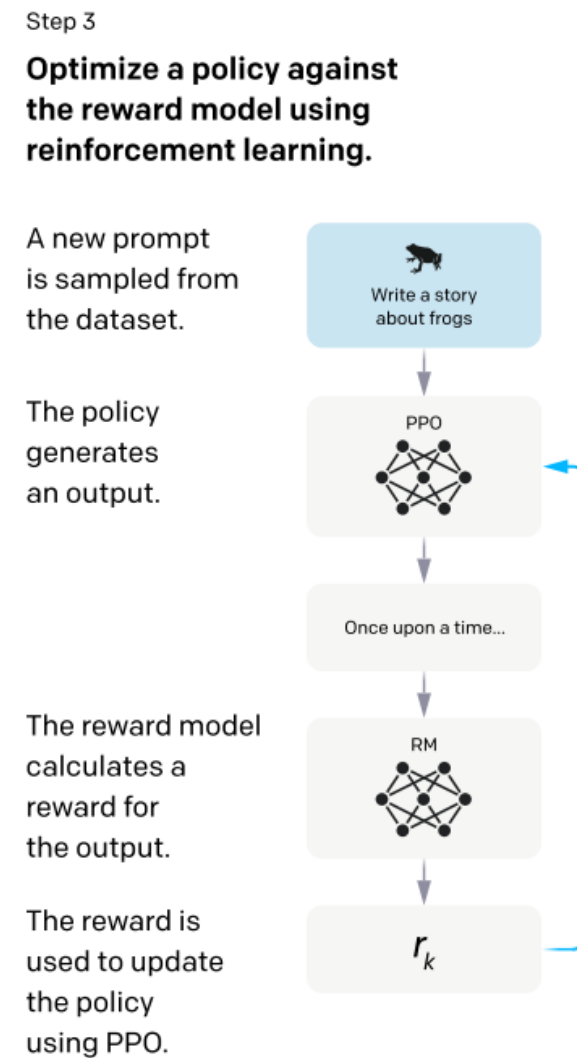
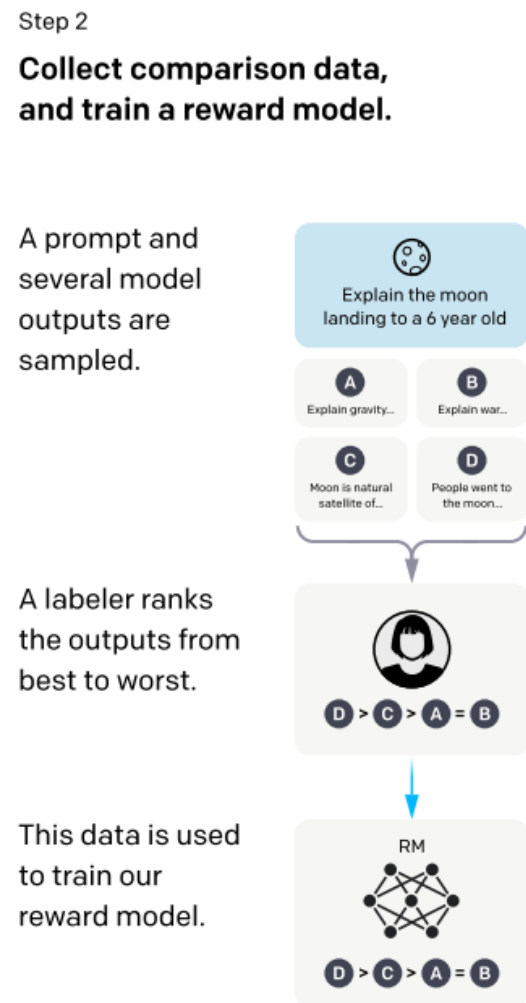
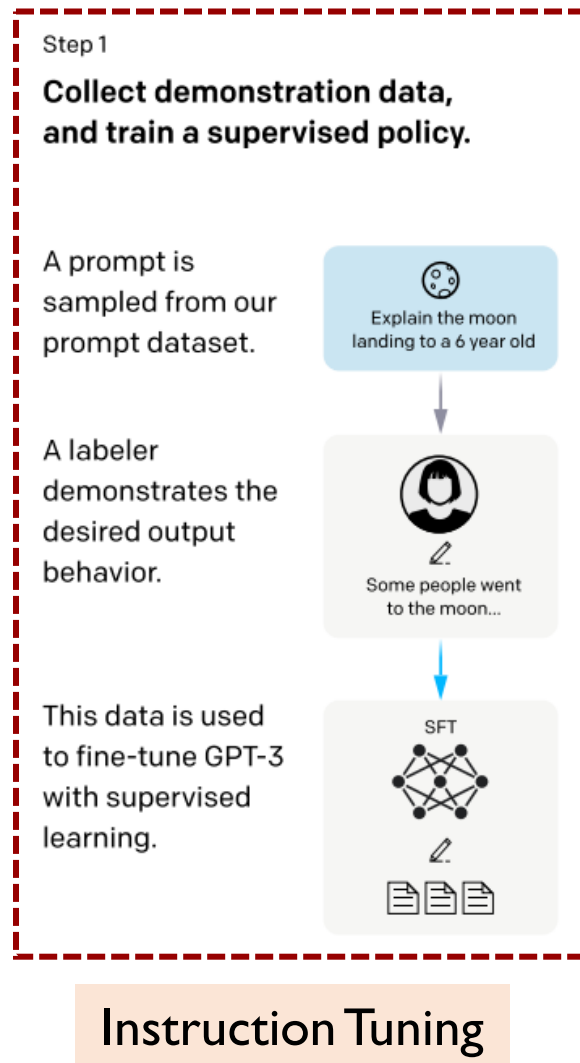
(d) FLAN

FLAN: 62 tasks



(e) INSTRUCTGPT

InstructGPT [Ouyang et al., 2022]

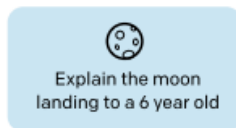


InstructGPT

Step 1

Collect demonstration data, and train a supervised policy.

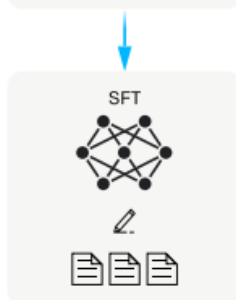
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



Train another model to imitate human preferences

Step 2

Collect comparison data, and train a reward model.

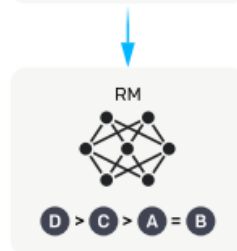
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Train GPT to satisfy the “human” model

Step 3

Optimize a policy against the reward model using reinforcement learning.

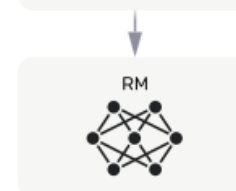
A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

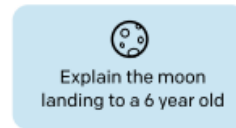


InstructGPT

Step 1

**Collect demonstration data,
and train a supervised policy.**

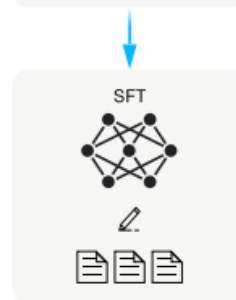
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



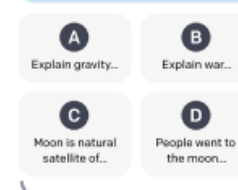
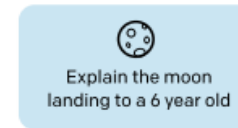
This data is used
to fine-tune GPT-3
with supervised
learning.



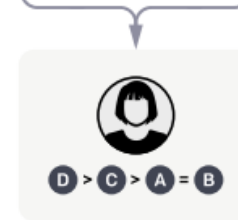
Step 2

**Collect comparison data,
and train a reward model.**

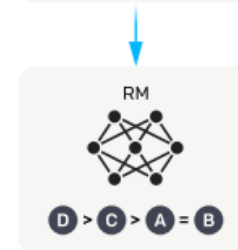
A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



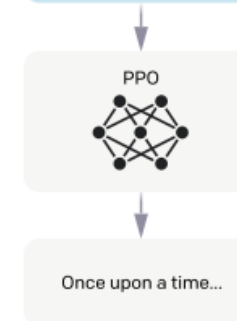
Step 3

**Optimize a policy against
the reward model using
reinforcement learning.**

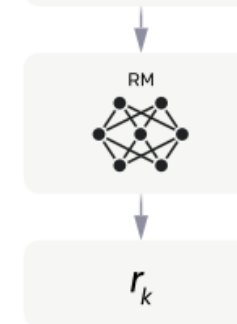
A new prompt
is sampled from
the dataset.



The policy
generates
an output.



The reward model
calculates a
reward for
the output.



The reward is
used to update
the policy
using PPO.



Zero-Shot CoT Prompting [Kojima et al., 2022]

(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

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. ✓

(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

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?

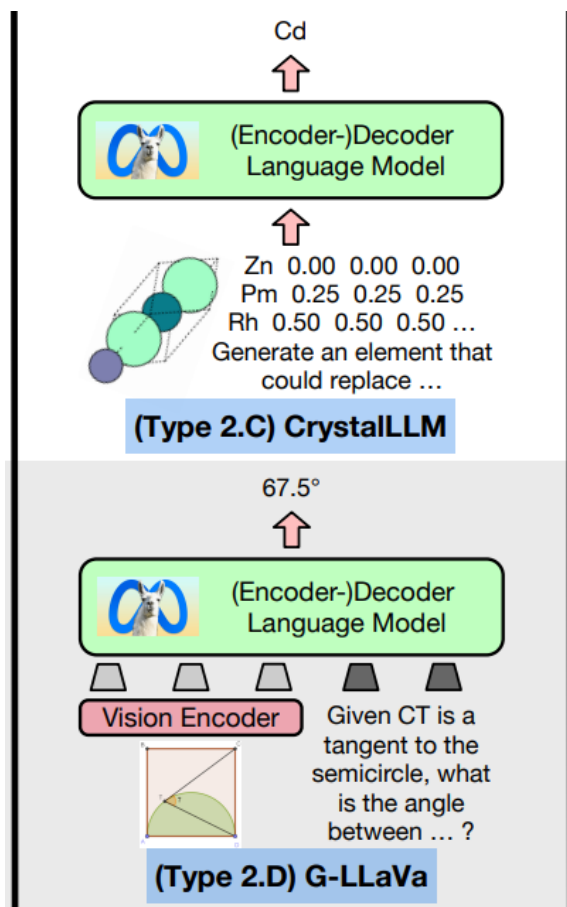
A: **Let's think step by step.**

(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. ✓

The Power of “*Let’s think step by step.*”

No.	Category	Template	Accuracy
1	instructive	Let’s think step by step.	78.7
2		First, (*1)	77.3
3		Let’s think about this logically.	74.5
4		Let’s solve this problem by splitting it into steps. (*2)	72.2
5		Let’s be realistic and think step by step.	70.8
6		Let’s think like a detective step by step.	70.3
7		Let’s think	57.5
8		Before we dive into the answer,	55.7
9		The answer is after the proof.	45.7
10	misleading	Don’t think. Just feel.	18.8
11		Let’s think step by step but reach an incorrect answer.	18.7
12		Let’s count the number of "a" in the question.	16.7
13		By using the fact that the earth is round,	9.3
14	irrelevant	By the way, I found a good restaurant nearby.	17.5
15		AbraKadabra!	15.5
16		It’s a beautiful day.	13.1
-		(Zero-shot)	17.7

A Survey of Scientific LLMs [Zhang et al., 2024]



The screenshot shows the GitHub repository page for "Awesome Scientific Language Models". The page includes a README, MIT license, and a curated list of pre-trained language models in scientific domains. The repository has 630 stars and 296 papers. The page also mentions that the repository is part of a survey paper and will be continuously updated.

Awesome Scientific Language Models

[awesome](#) [Stars](#) 630


[PaperNumber](#) 296 [License](#) MIT [PRs](#) Welcome

A curated list of pre-trained language models in scientific domains (e.g., mathematics, physics, chemistry, materials science, biology, medicine, geoscience), covering different model sizes (from 100M to 100B parameters) and modalities (e.g., language, graph, vision, table, molecule, protein, genome, climate time series).


The repository is part of our survey paper [A Comprehensive Survey of Scientific Large Language Models and Their Applications in Scientific Discovery](#) and will be continuously updated.

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**.

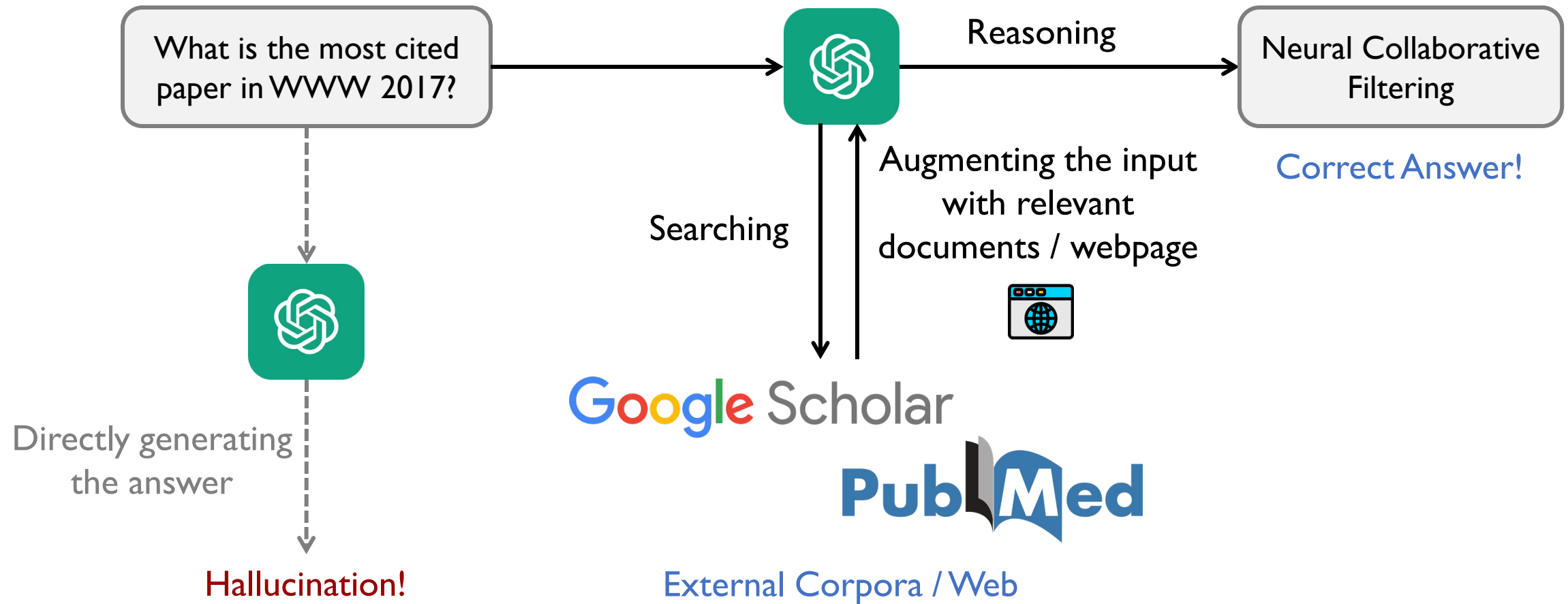
HIN2Vec was written by **Tao-yang Fu, Wang-Chien Lee, and Zhen Lei**.

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
- Solutions?
 - Teach the LLM to *know one more fact* (e.g., “*HIN2Vec was published in CIKM 2017.*”)
 - Tomorrow there will be yet another new conference proceedings. What should we do when people start asking questions about this conference?
 - Teach the LLM to *explore the external corpora/Web*
 - It can leverage the freshness of the Web to handle a continuous stream of new questions.

Retrieval-Augmented Generation (RAG) [Lewis et al., 2020]



Search Engine Basics

Query: describing the information we are seeking



What is the most cited paper in WWW 2017?



Massachusetts Institute of Technology

[https://direct.mit.edu › rest › article › Most-Cited-Papers...](https://direct.mit.edu/rest/article/Most-Cited-Papers...)

Most Cited Papers: 1919–2017

Mar 1, 2018 — **Most Cited Papers: 1919–2017** Available ... The Review of Economics and Statistics (2018) 100 (1): i–vii. ... This content is only available as a PDF ... [Read more](#)



National Institutes of Health (NIH) | (.gov)

[https://pmc.ncbi.nlm.nih.gov › articles › PMC8551947](https://pmc.ncbi.nlm.nih.gov/articles/PMC8551947)

The 50 Most Highly Cited Reviews of 2013–2017 - PMC

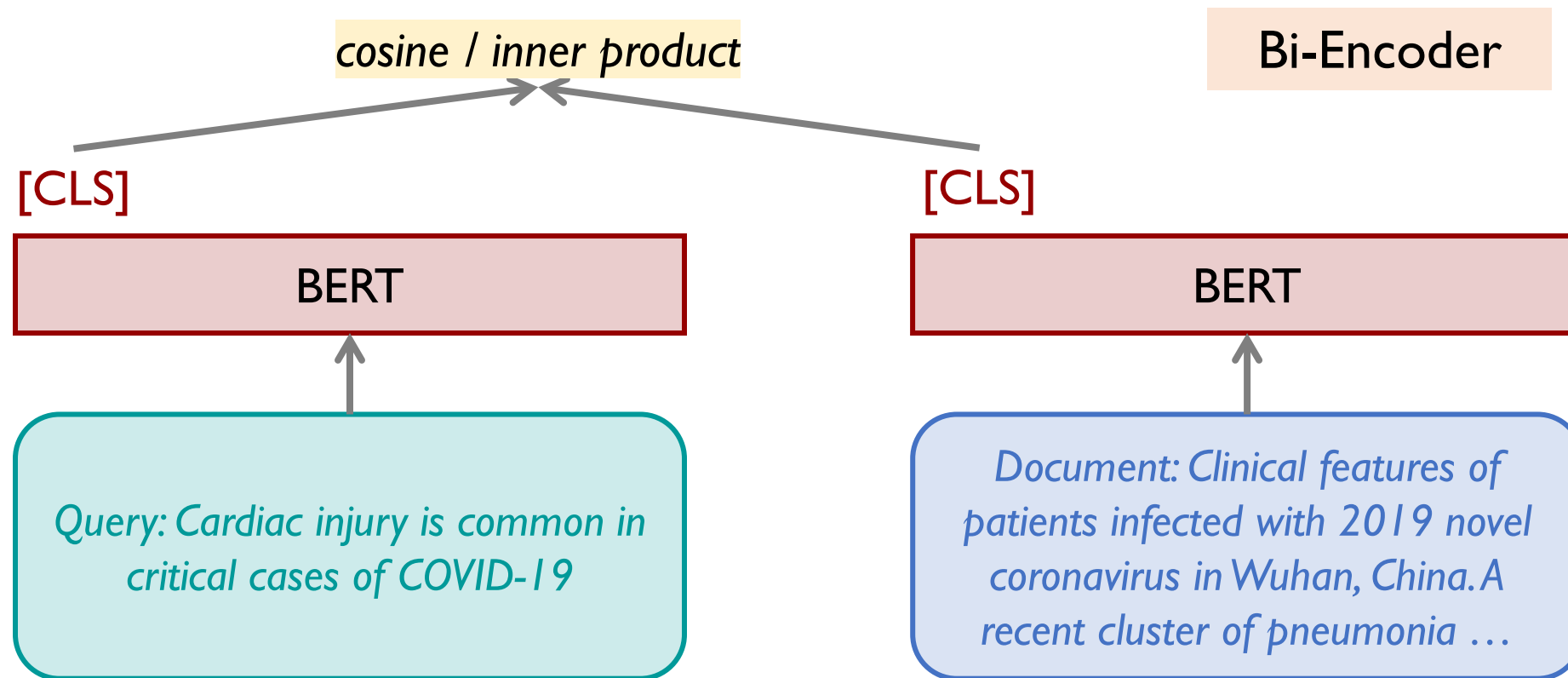
by AV Bagirova · 2021 · Cited by 6 — The main characteristics of the 50 most highly cited reviews based on Scopus data, published in 2013–2017, have been studied. [Read more](#)



Documents: the data collection that the system searches through; may contain millions/billions of candidates

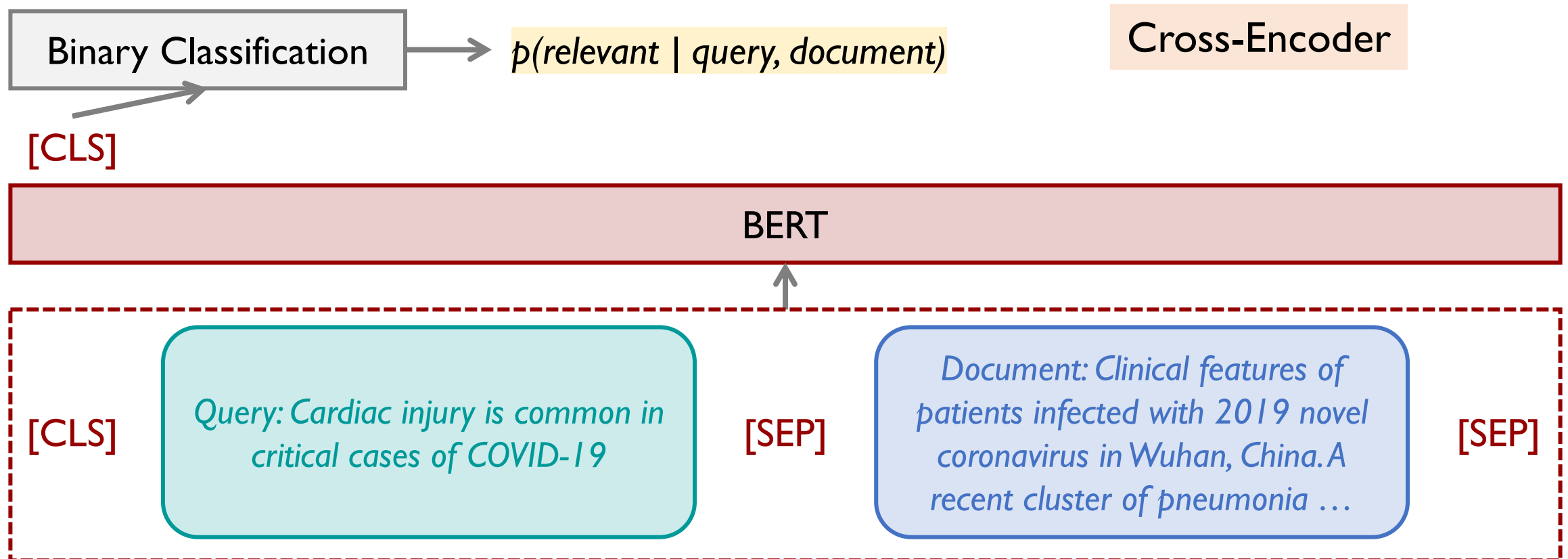
Using LLMs (e.g., BERT) for Search – Solution 1

- Encode **query** and **document** separately
- The output vector of the [CLS] token serves as **query** / **document** embedding



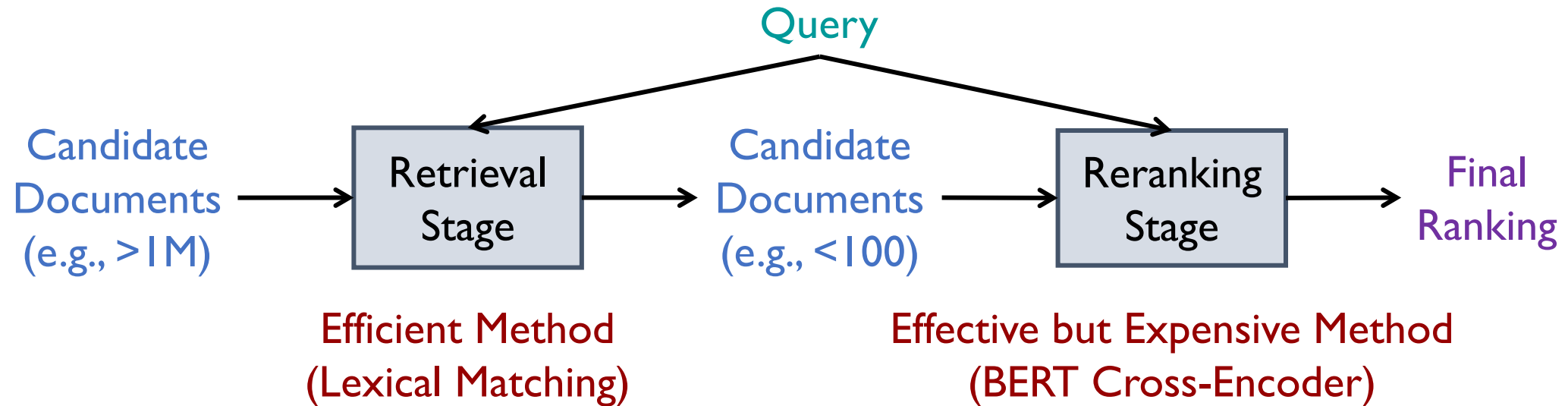
Using LLMs (e.g., BERT) for Search – Solution 2

- Concatenate the **query** and **document** into a single input sequence
- Get the representation of the entire sequence and perform binary classification



The Retrieval-Reranking Paradigm for Search

- We want to use effective but expensive ranking models ...
- ... only for a more fine-grained ranking of the most relevant documents.



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 [ARXIV](#) [NEURIPS PROCEEDINGS](#).



ChatGPT 4o, queried on October 12, 2024

Transformer was published
in **NeurIPS 2017**.

Toolformer [Schick et al., 2023]

- 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
- In-context few-shot learning
 - Instruction + demo examples

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., 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](#)

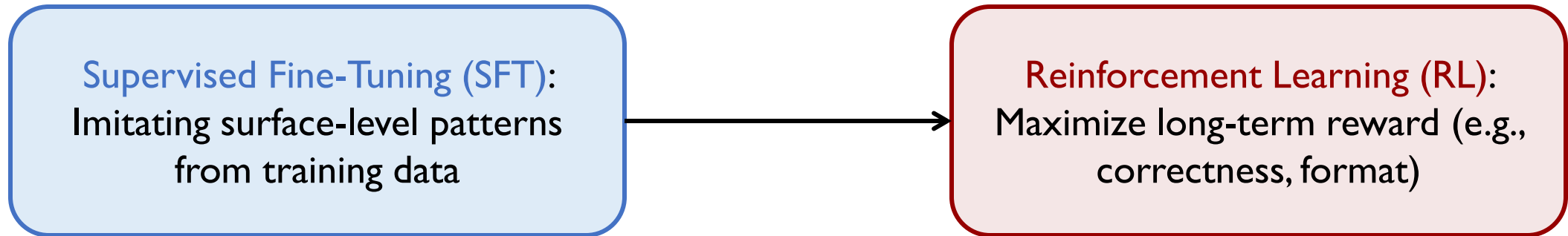
[Nature](#) **645**, 633–638 (2025) | [Cite this article](#)

278k Accesses | **53** Citations | **789** Altmetric | [Metrics](#)

Abstract

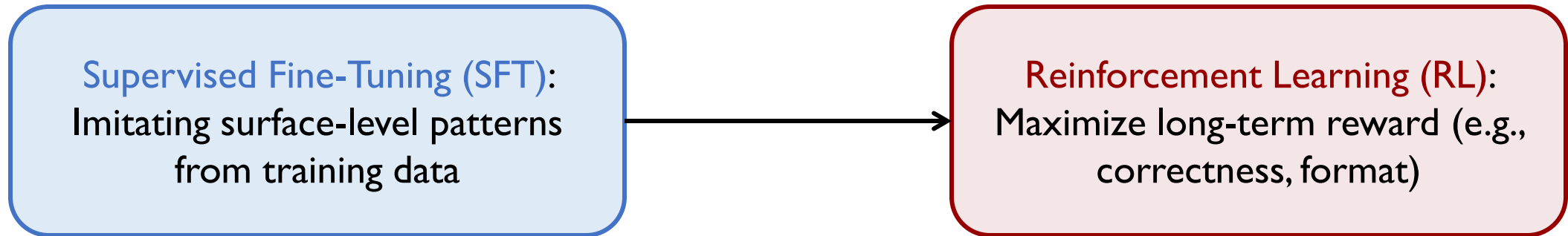
General reasoning represents a long-standing and formidable challenge in artificial intelligence (AI). Recent breakthroughs, exemplified by large language models (LLMs)^{1,2} and

How to force LLMs to reason?



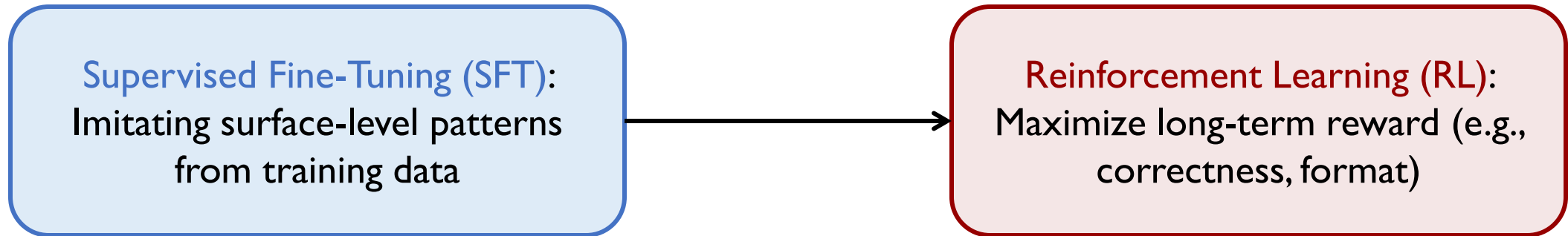
- **Supervised Fine-Tuning (SFT):** Given a question, maximize the likelihood that the LLM generates the reference answer.
 - If the reference answer is 0.5, then generating 1/2 is wrong.
 - If the LLM is asked to write code for a specific function, only an exact match with the reference answer is considered 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?



- **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?



- **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.

More about RL, Reasoning, and Agents?



Thursday (January 8):

Day 4 — Reinforcement Learning, Post-Training, Reasoning, and Language Agents

- [Dileep Kalathil](#), Associate Professor, Electrical & Computer Engineering
- [Tianbao Yang](#), Professor, Computer Science & Engineering

Thank You!