# CSCI 5541: Natural Language Processing

## Lecture 13: Instructing and augmenting LLMs

Slides are made by Zae Myung Kim (https://zaemyung.github.io/)

computer science
& engineering

MINNESOTA · NLP · EST. 2021

UNIVERSITY OF MINNESOTA
Driven to Discover®

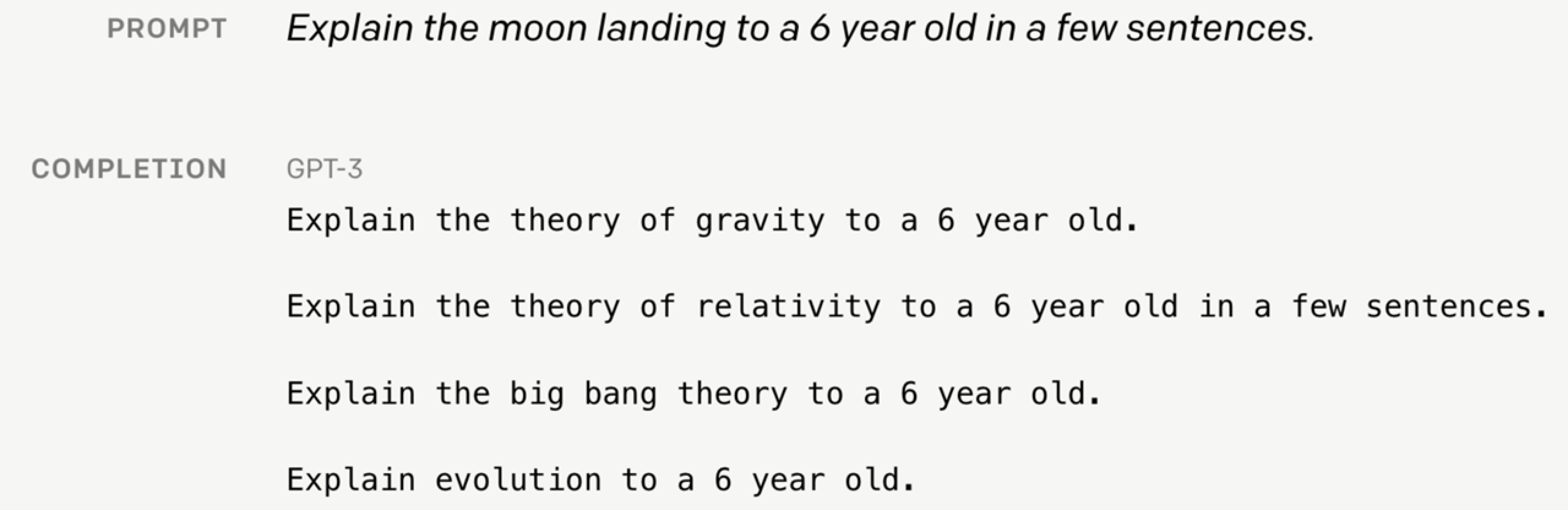
# Announcements

- ❑ Proposal Reports in (please review our comments in slack)
- ❑ Midterm Office Hour reminder (Apr 8)
- ❑ HW4 → Due this Thursday (Mar 27)

# Agenda

- Instruction Tuning
- Augmented Language Models
- Reasoning
- Using Tools and Act
- Evaluation

# Instruction Tuning



**PROMPT** *Explain the moon landing to a 6 year old in a few sentences.*

**COMPLETION** GPT-3

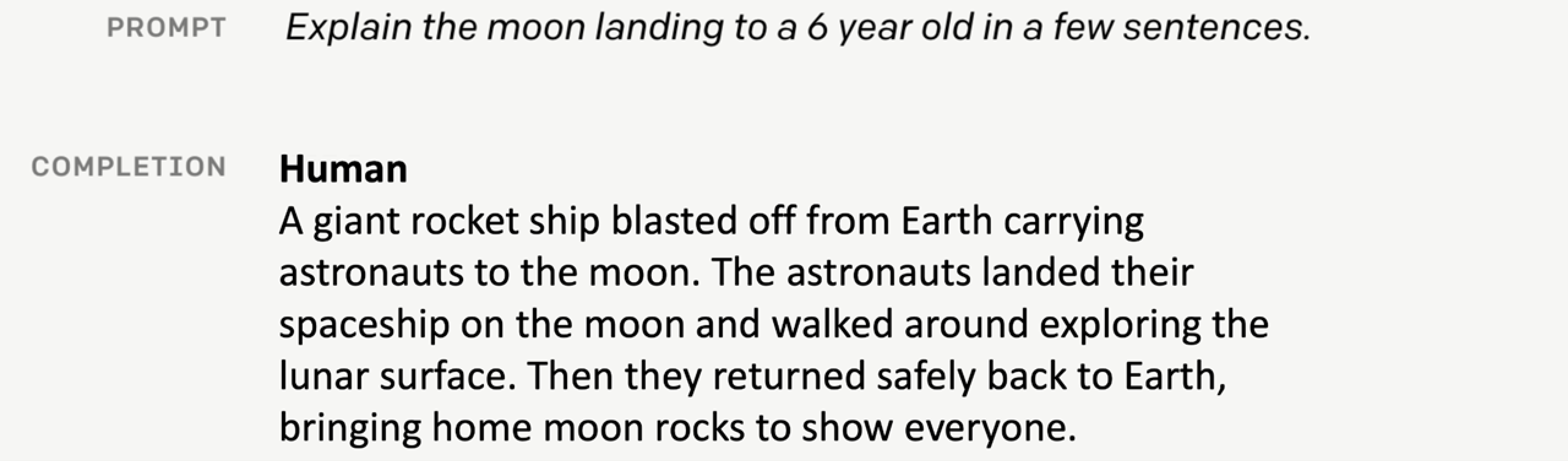
```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Language models are not aligned with user intent [Ouyang et al., 2022].



**PROMPT** *Explain the moon landing to a 6 year old in a few sentences.*

**COMPLETION**

**Human**

A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

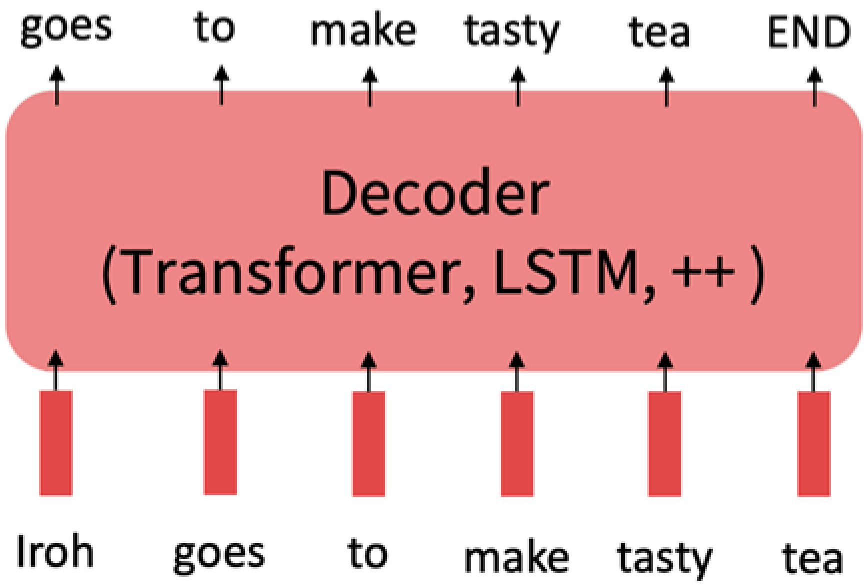
Language models are not aligned with user intent [Ouyang et al., 2022].

→ We can *finetune* it with responses we want!

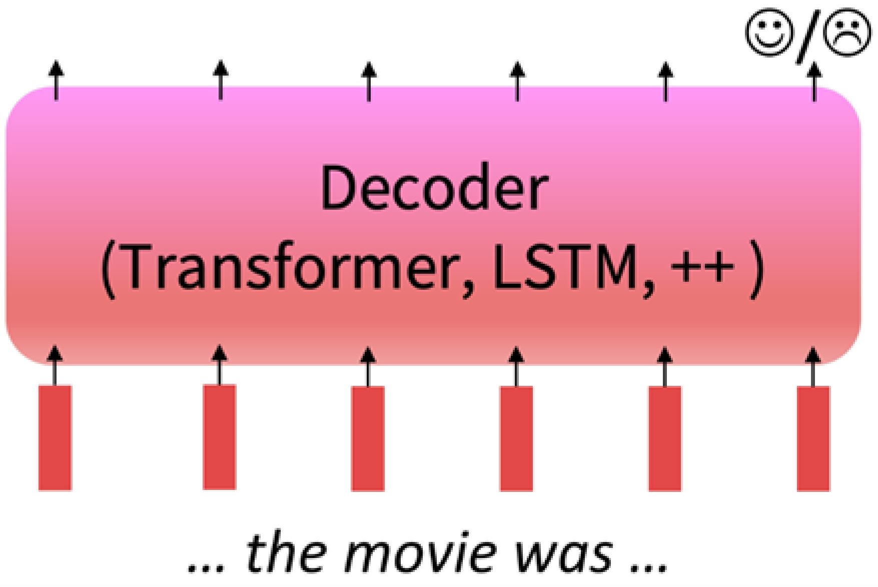
# Recap on pretrain-finetune paradigm

- ☐ Pretraining can greatly improve performances on downstream NLP tasks by serving as parameter initialization.

**Step 1: Pretrain (via language modeling)**
Lots of text; learn general things!



**Step 2: Finetune (on your task)**
Not many labels; adapt to the task!



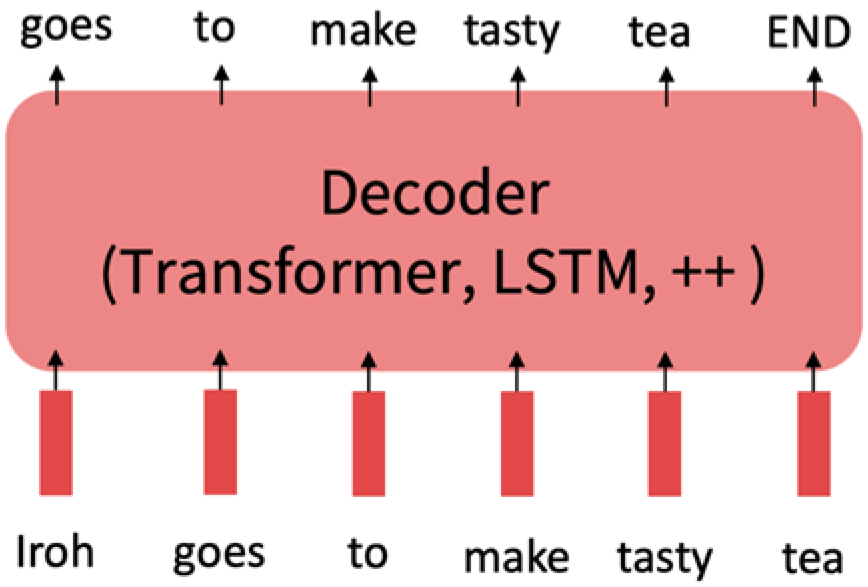
Source: cs224n, Stanford

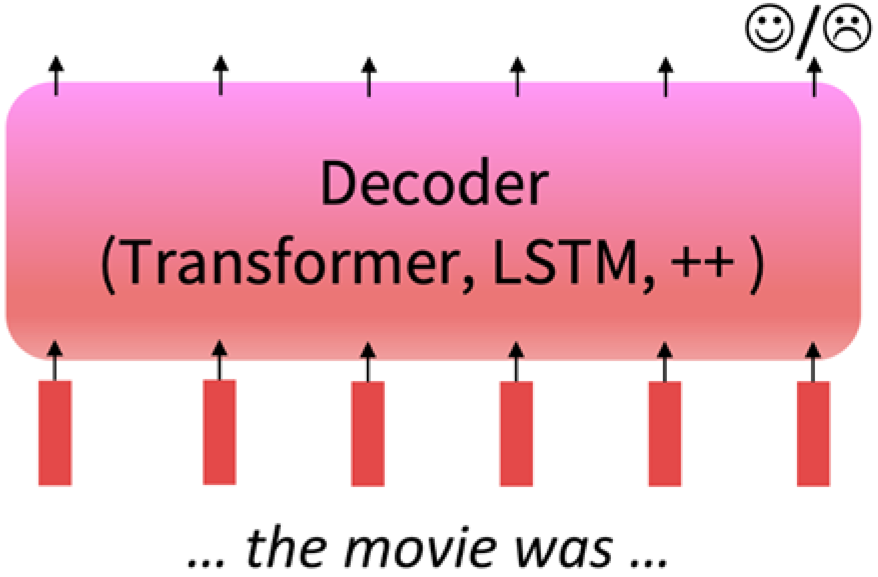
# Scaling up finetuning

- ☐ Pretraining can greatly improve performances on downstream NLP tasks by serving as parameter initialization.

**Step 1: Pretrain (via language modeling)**
Lots of text; learn general things!

**Step 2: Finetune (on many tasks)**
Many labels; adapt to many tasks!





Source: cs224n, Stanford

# How does prompting fit into this paradigm?

> Please label the sentiment towards the movie of the given movie review. The sentiment label should be "positive" or "negative".
> Text: i'll bet the video game is a lot more fun than the film.
> Sentiment:

For example to produce education materials for kids,

> Describe what is quantum physics to a 6-year-old.

And safe content,

> … in language that is safe for work.

Please label the sentiment towards the movie of the given movie review. The sentiment label should be "positive" or "negative".

Text: i'll bet the video game is a lot more fun than the film.

Sentiment:

For example to produce education materials for kids,

Describe what is quantum physics to a 6-year-old.

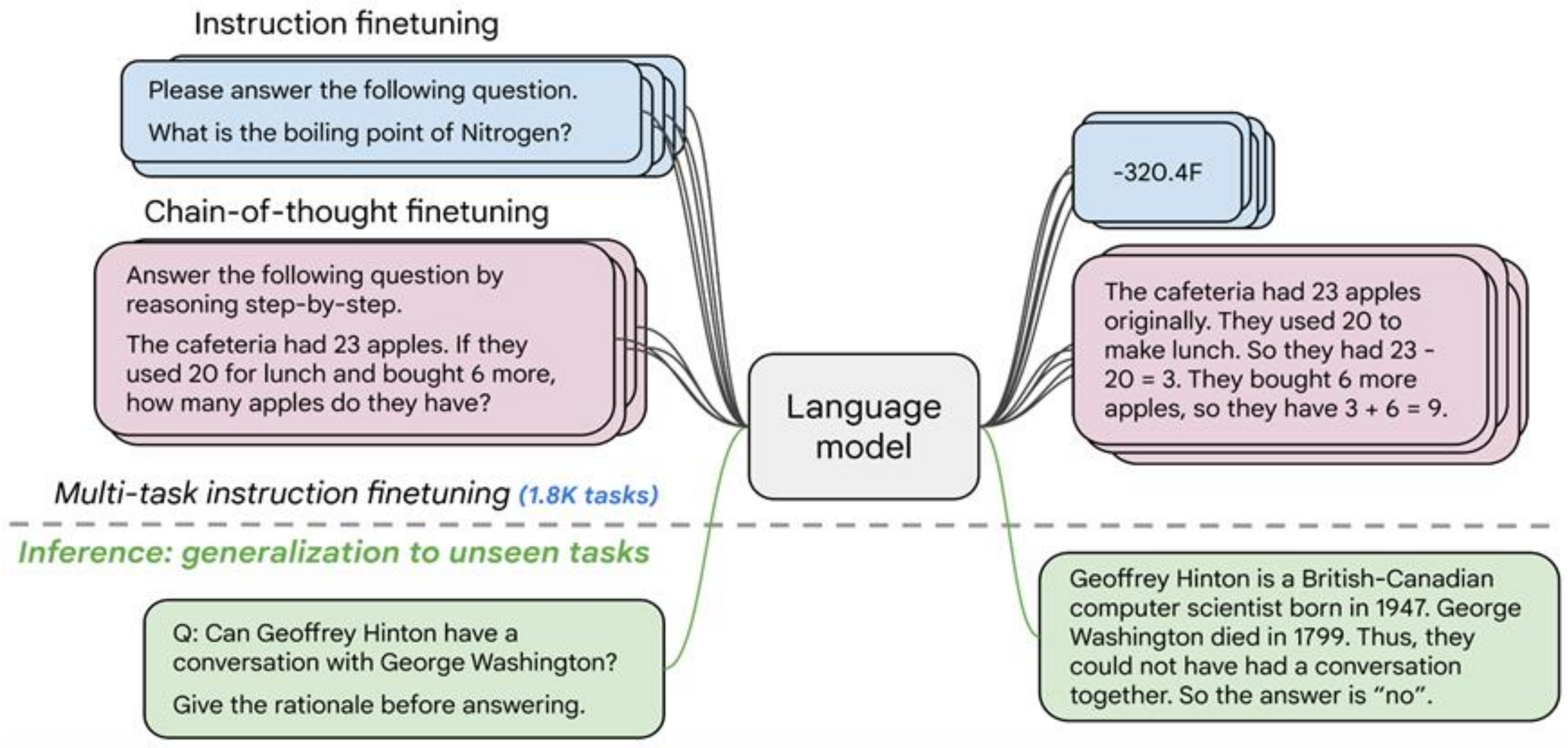
And safe content,

... in language that is safe for work.

Train on your prompts

# Instruction finetuning

- Collect examples of ((instruction, input), output) pairs across many tasks and finetune an LM and evaluate on unseen tasks


Instruction finetuning
Chain-of-thought finetuning
Multi-task instruction finetuning (1.8K tasks)
Inference: generalization to unseen tasks
Language model

Please answer the following question.
What is the boiling point of Nitrogen?

-320.4F

Answer the following question by reasoning step-by-step.
The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

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

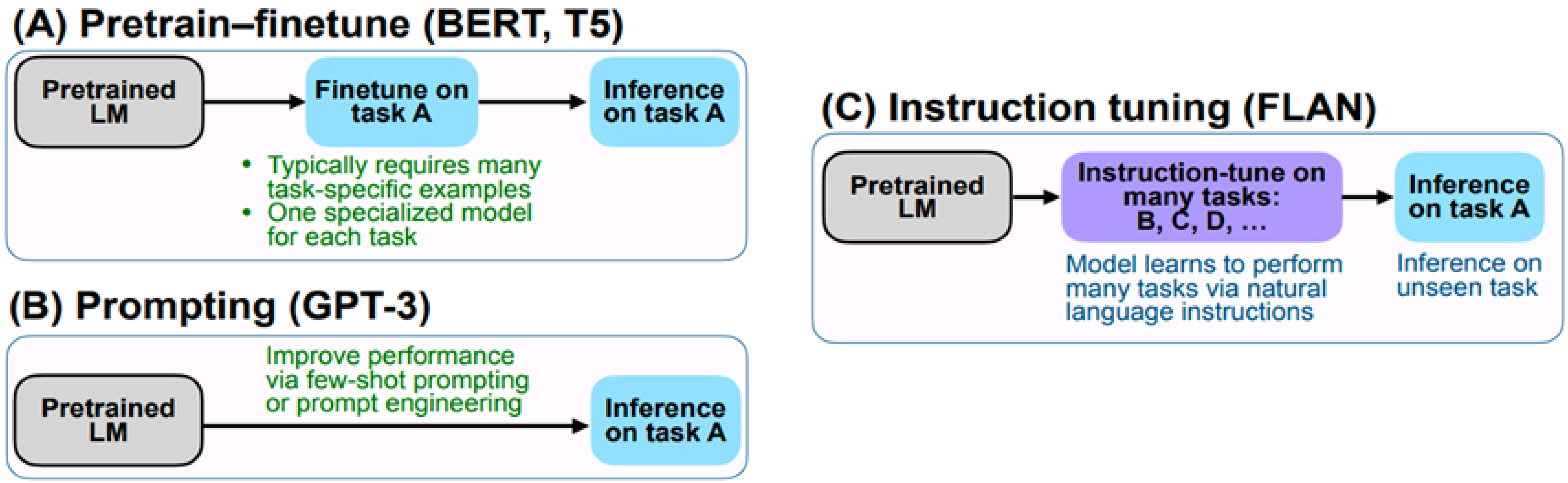
Q: Can Geoffrey Hinton have a conversation with George Washington?
Give the rationale before answering.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is "no".

[FLAN-T5; Chung et al., 2022]

# Instruction finetuning vs. standard finetuning

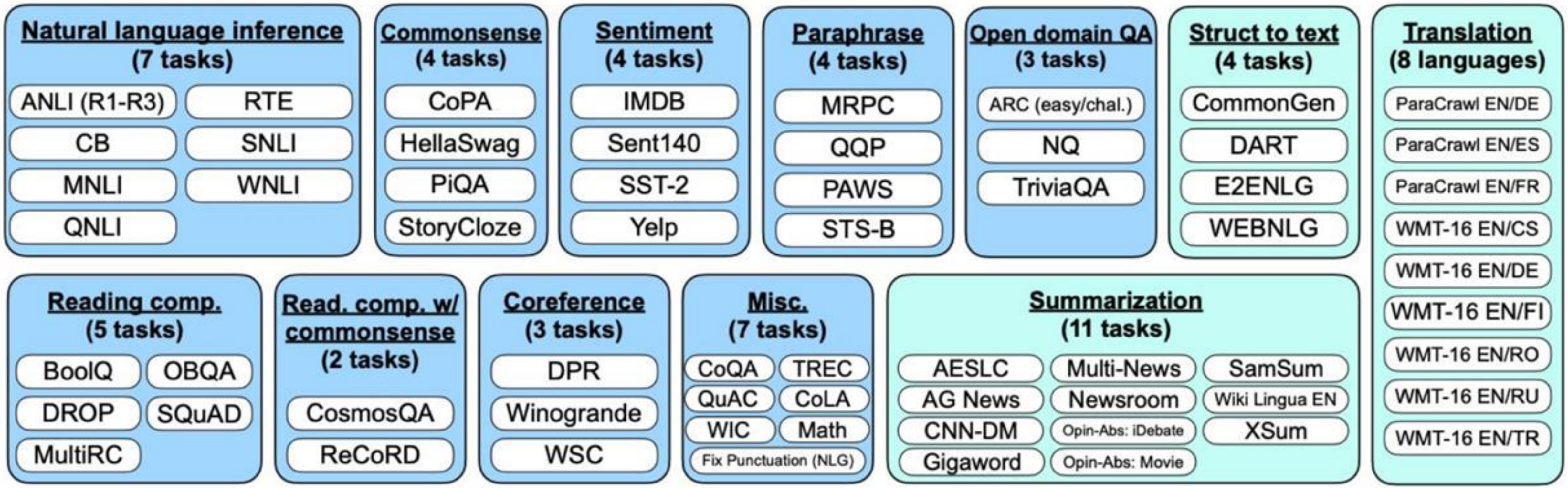
- ❑ The main difference lies in the data that the model is trained on
  - ○ Standard supervised finetuning trains models on ***input examples*** and their ***corresponding outputs***.
  - ○ Instruction finetuning augments ***input-output examples*** with ***instructions***, which enables instruction-tuned models to generalize more easily to new tasks.


(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

[Wei et al., 2022]

# Instruction finetuning (Early Example)

- ❑ 62 NLP datasets
- ❑ 12 “task clusters”



**Natural language inference (7 tasks)**
- ANLI (R1-R3)
- RTE
- CB
- SNLI
- MNLI
- WNLI
- QNLI

**Commonsense (4 tasks)**
- CoPA
- HellaSwag
- PiQA
- StoryCloze

**Sentiment (4 tasks)**
- IMDB
- Sent140
- SST-2
- Yelp

**Paraphrase (4 tasks)**
- MRPC
- QQP
- PAWS
- STS-B

**Open domain QA (3 tasks)**
- ARC (easy/chal.)
- NQ
- TriviaQA

**Struct to text (4 tasks)**
- CommonGen
- DART
- E2ENLG
- WEBNLG

**Translation (8 languages)**
- ParaCrawl EN/DE
- ParaCrawl EN/ES
- ParaCrawl EN/FR
- WMT-16 EN/CS
- WMT-16 EN/DE
- WMT-16 EN/FI
- WMT-16 EN/RO
- WMT-16 EN/RU
- WMT-16 EN/TR

**Reading comp. (5 tasks)**
- BoolQ
- OBQA
- DROP
- SQuAD
- MultiRC

**Read. comp. w/ commonsense (2 tasks)**
- CosmosQA
- ReCoRD

**Coreference (3 tasks)**
- DPR
- Winogrande
- WSC

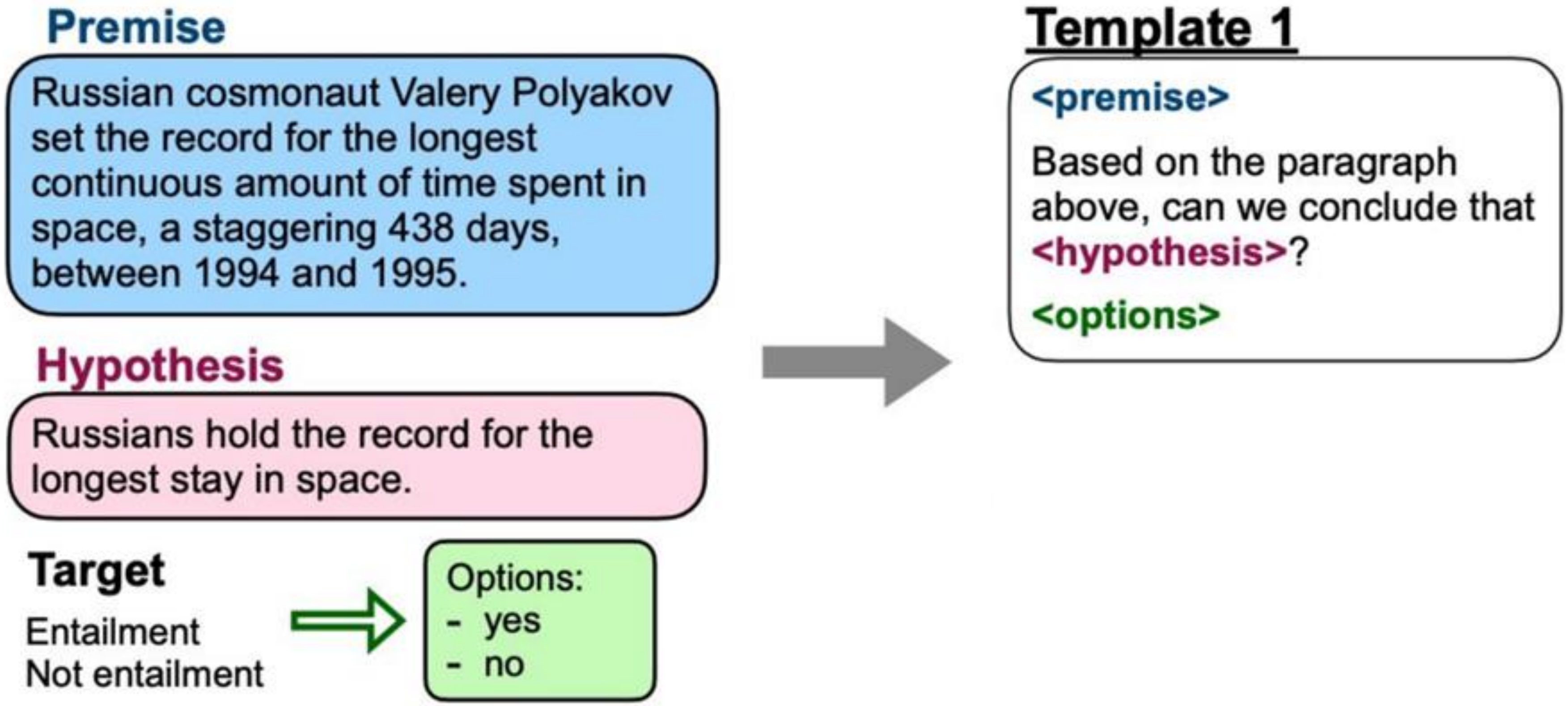
**Misc. (7 tasks)**
- CoQA
- TREC
- QuAC
- CoLA
- WIC
- Math
- Fix Punctuation (NLG)

**Summarization (11 tasks)**
- AESLC
- Multi-News
- SamSum
- AG News
- Newsroom
- Wiki Lingua EN
- CNN-DM
- Opin-Abs: iDebate
- XSum
- Gigaword
- Opin-Abs: Movie

[FLAN-T5; Chung et al., 2022]

# Instruction finetuning templates

- ❑ Natural instruction templates for each task



**Premise**

Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

**Hypothesis**

Russians hold the record for the longest stay in space.

**Target**

Entailment
Not entailment

Options:
- yes
- no

**Template 1**

<premise>

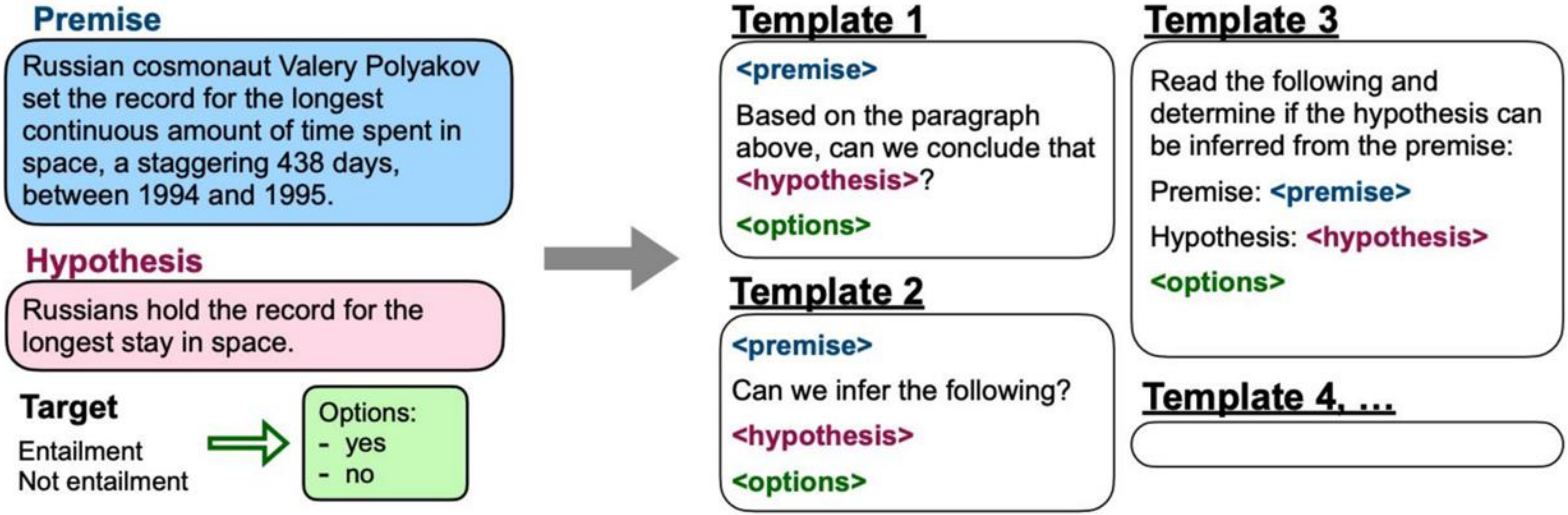
Based on the paragraph above, can we conclude that <hypothesis>?

<options>

[FLAN-T5; Chung et al., 2022]

# Instruction finetuning templates

- ☐ Natural instruction templates for each task



**Premise**

Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

**Hypothesis**

Russians hold the record for the longest stay in space.

**Target**

Entailment
Not entailment

Options:
- yes
- no

**Template 1**

<premise>

Based on the paragraph above, can we conclude that <hypothesis>?

<options>

**Template 2**

<premise>

Can we infer the following?

<hypothesis>

<options>

**Template 3**

Read the following and determine if the hypothesis can be inferred from the premise:

Premise: <premise>

Hypothesis: <hypothesis>

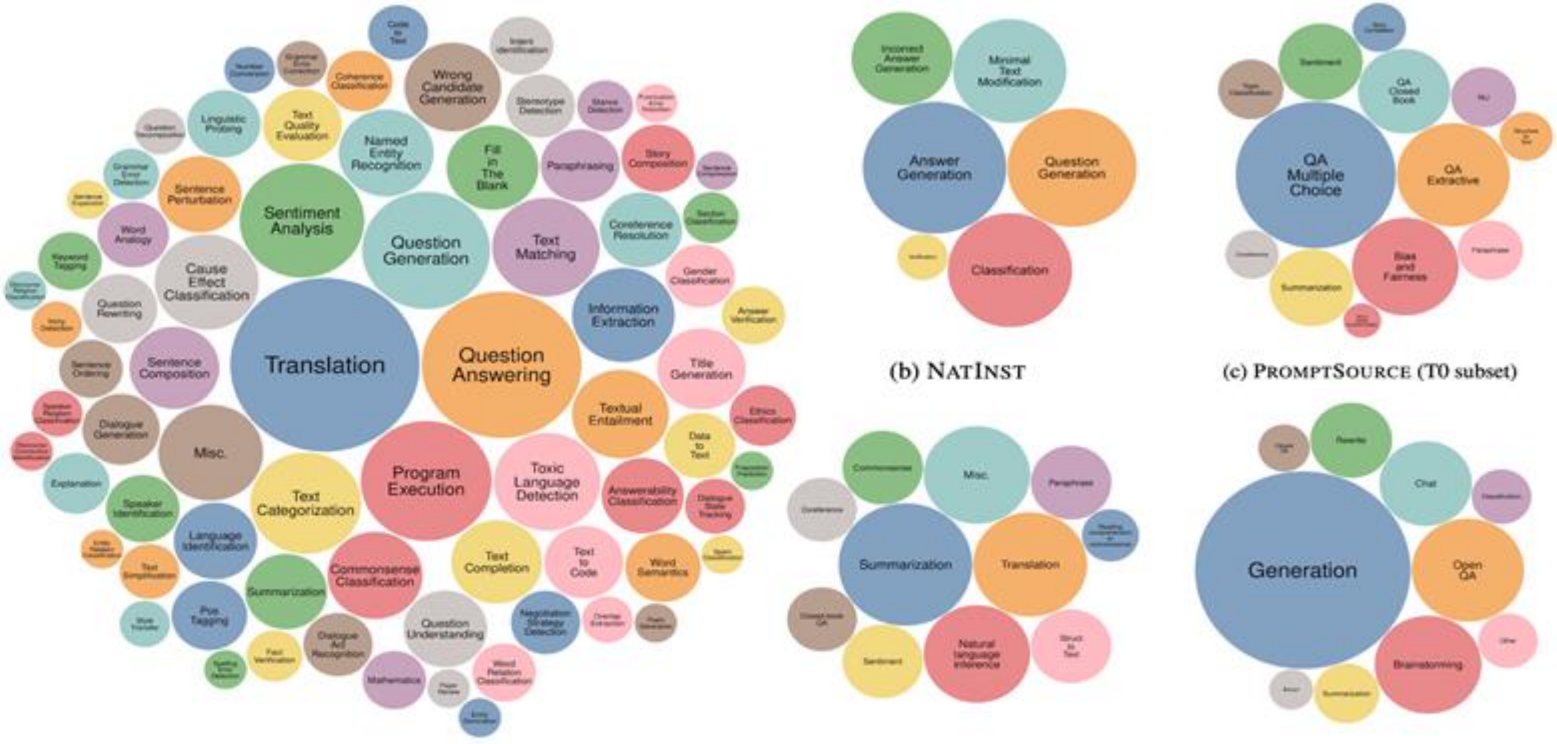
<options>

**Template 4, …**

[FLAN-T5; Chung et al., 2022]

# Instruction pretraining?

- ❑ Scaling up data and model improves performance.
- ❑ **SuperNaturalInstructions** dataset contains **over 1.6K tasks, 3M+ examples**



(a) SUP-NATINST (this work)

(b) NATINST

(c) PROMPTSOURCE (T0 subset)

(d) FLAN

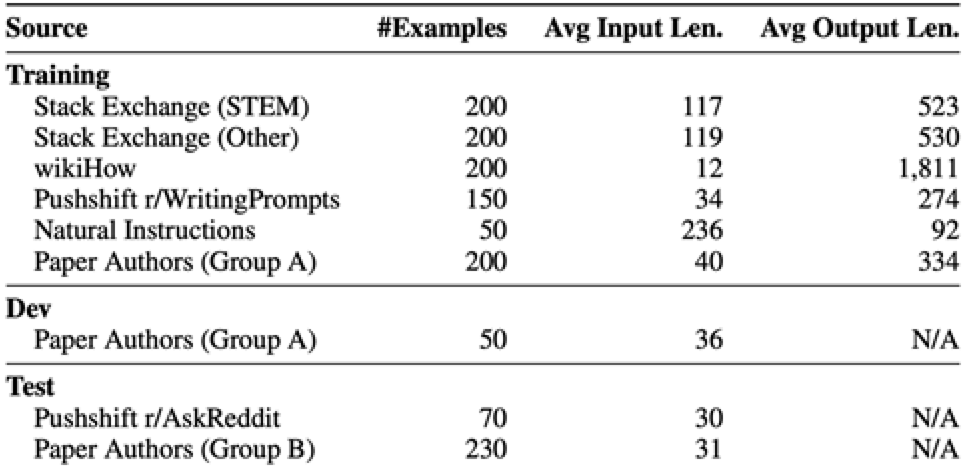
(e) INSTRUCTGPT

[Wang et al., 2022]

# Or less is more?

*"LIMA – Less Is More for Alignment"* ([Zhou et al. 2023](#))

- Authors report that LLaMa 65B model finetuned on a collection of high quality and diverse 1,000 samples are enough to beat models trained on much larger instruction datasets.



| Source | #Examples | Avg Input Len. | Avg Output Len. |
|---|---|---|---|
| **Training** | | | |
| Stack Exchange (STEM) | 200 | 117 | 523 |
| Stack Exchange (Other) | 200 | 119 | 530 |
| wikiHow | 200 | 12 | 1,811 |
| Pushshift r/WritingPrompts | 150 | 34 | 274 |
| Natural Instructions | 50 | 236 | 92 |
| Paper Authors (Group A) | 200 | 40 | 334 |
| **Dev** | | | |
| Paper Authors (Group A) | 50 | 36 | N/A |
| **Test** | | | |
| Pushshift r/AskReddit | 70 | 30 | N/A |
| Paper Authors (Group B) | 230 | 31 | N/A |

Table 1: Sources of training prompts (inputs) and responses (outputs), and test prompts. The total amount of training data is roughly 750,000 tokens, split over exactly 1,000 sequences.

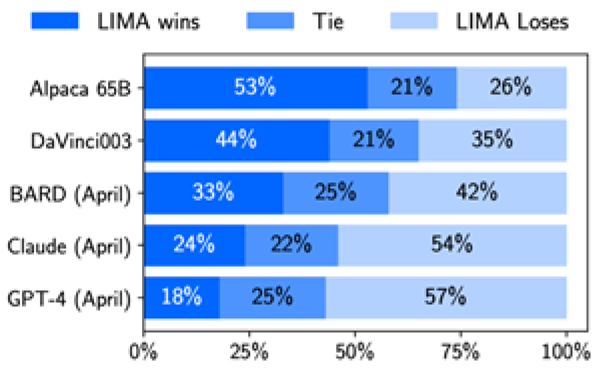


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

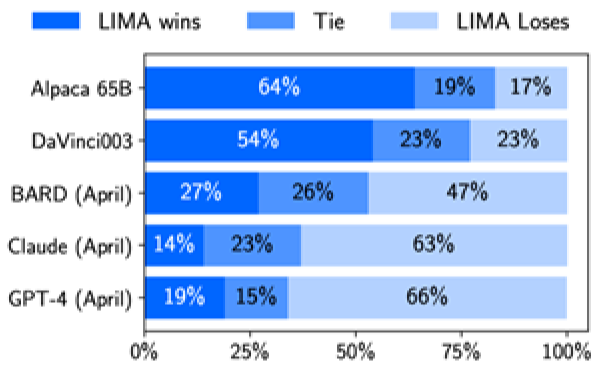
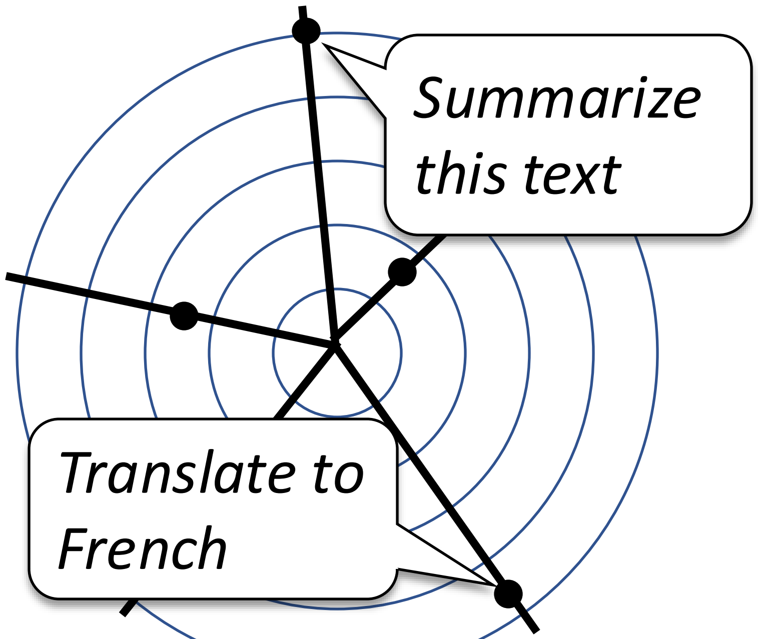
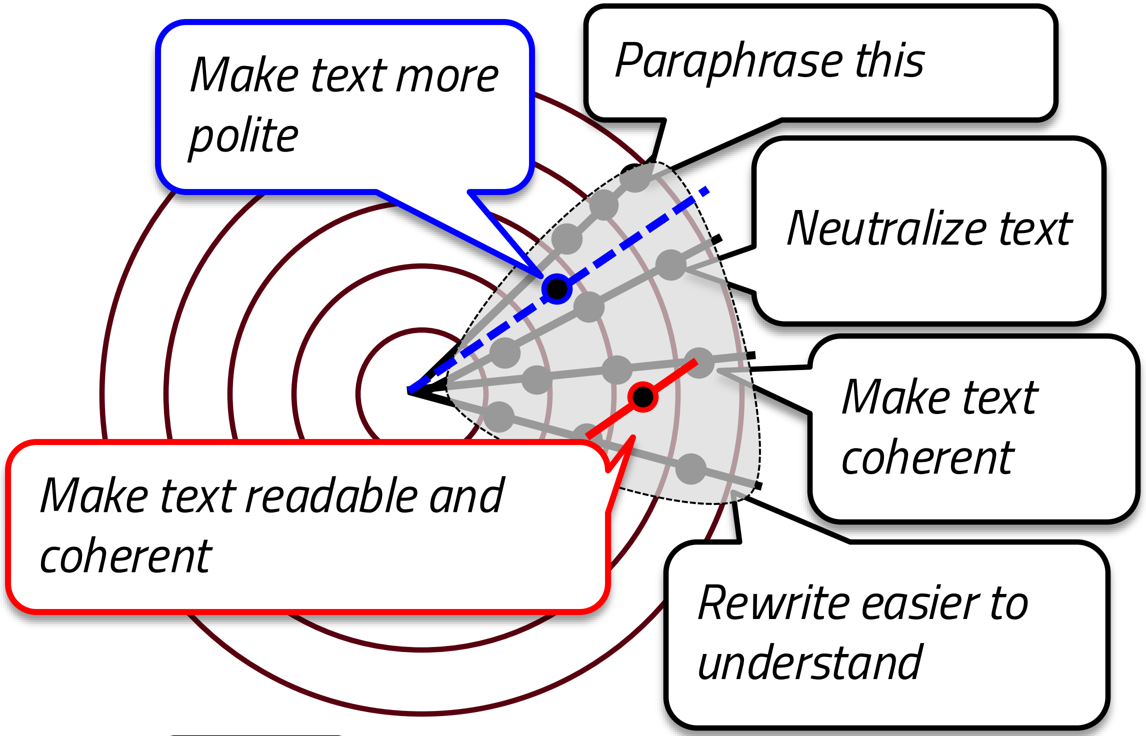


Figure 2: Preference evaluation using GPT-4 as the annotator, given the same instructions provided to humans.

# General-purpose (left) vs Task-specific (right) Instruction Tuning



*Sparse* task distribution



*Dense* task distribution

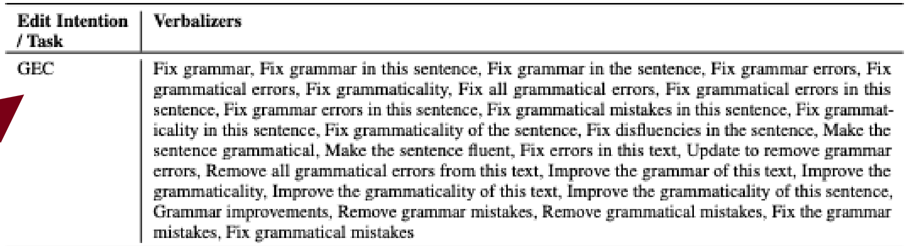
*Generalize to composite & unseen tasks*

Densifying the task distribution to strictly instructions within the text revision domain can largely improve model performance for revision tasks over scaling model size with general instructions

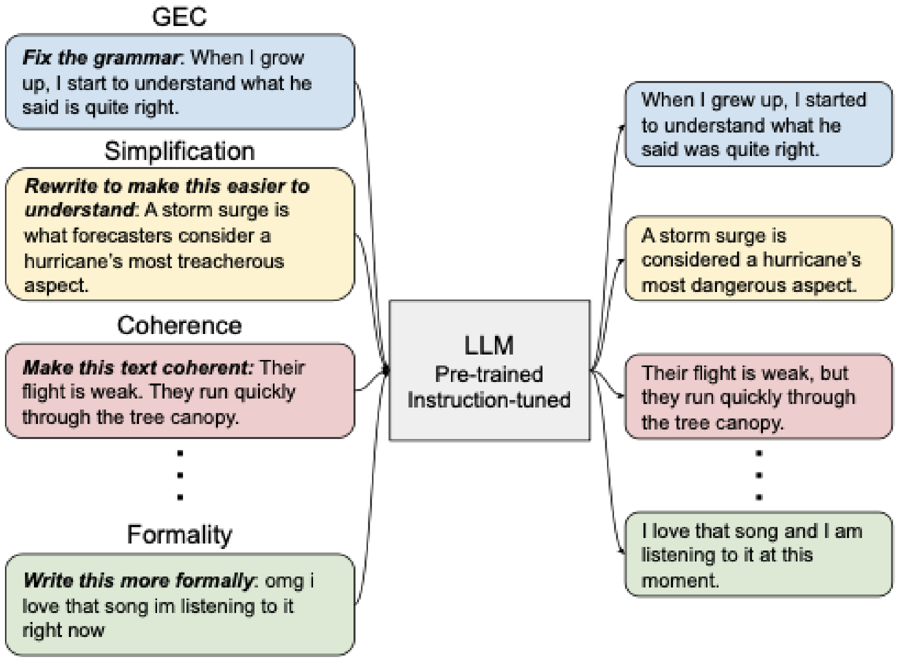
[Raheja et al., EMNLP Findings 23]

# User-driven Revision (CoEdIT)

Conversational Text Editing via Instruction Tuning



| Edit Intention / Task | Verbalizers |
|---|---|
| GEC | Fix grammar, Fix grammar in this sentence, Fix grammar in the sentence, Fix grammar errors, Fix grammatical errors, Fix grammaticality, Fix all grammatical errors, Fix grammatical errors in this sentence, Fix grammar errors in this sentence, Fix grammatical mistakes in this sentence, Fix grammaticality in this sentence, Fix grammaticality of the sentence, Fix disfluencies in the sentence, Make the sentence grammatical, Make the sentence fluent, Fix errors in this text, Update to remove grammar errors, Remove all grammatical errors from this text, Improve the grammar of this text, Improve the grammaticality, Improve the grammaticality of this text, Improve the grammaticality of this sentence, Grammar improvements, Remove grammar mistakes, Remove grammatical mistakes, Fix the grammar mistakes, Fix grammatical mistakes |



GEC

***Fix the grammar***: When I grow up, I start to understand what he said is quite right.

Simplification

***Rewrite to make this easier to understand***: A storm surge is what forecasters consider a hurricane's most treacherous aspect.

Coherence

***Make this text coherent:*** Their flight is weak. They run quickly through the tree canopy.

⋮

Formality

***Write this more formally***: omg i love that song im listening to it right now

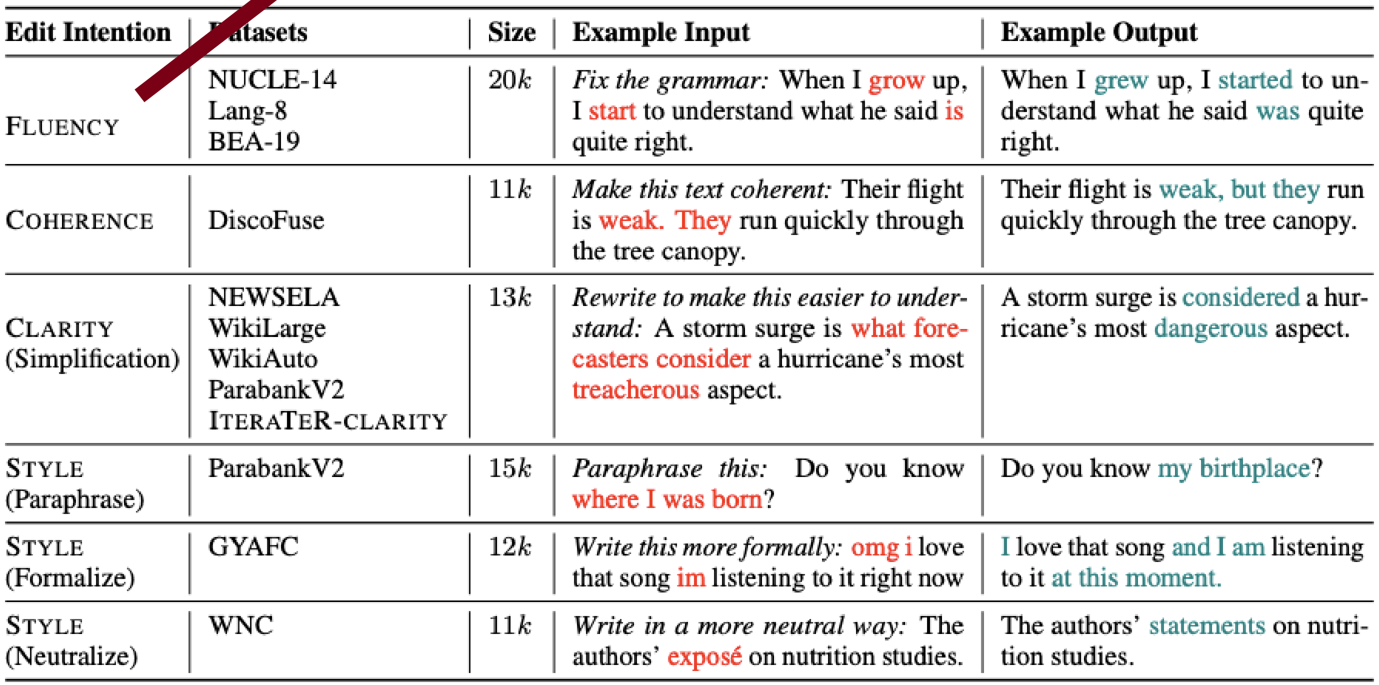
When I grew up, I started to understand what he said was quite right.

A storm surge is considered a hurricane's most dangerous aspect.

Their flight is weak, but they run quickly through the tree canopy.

⋮

I love that song and I am listening to it at this moment.

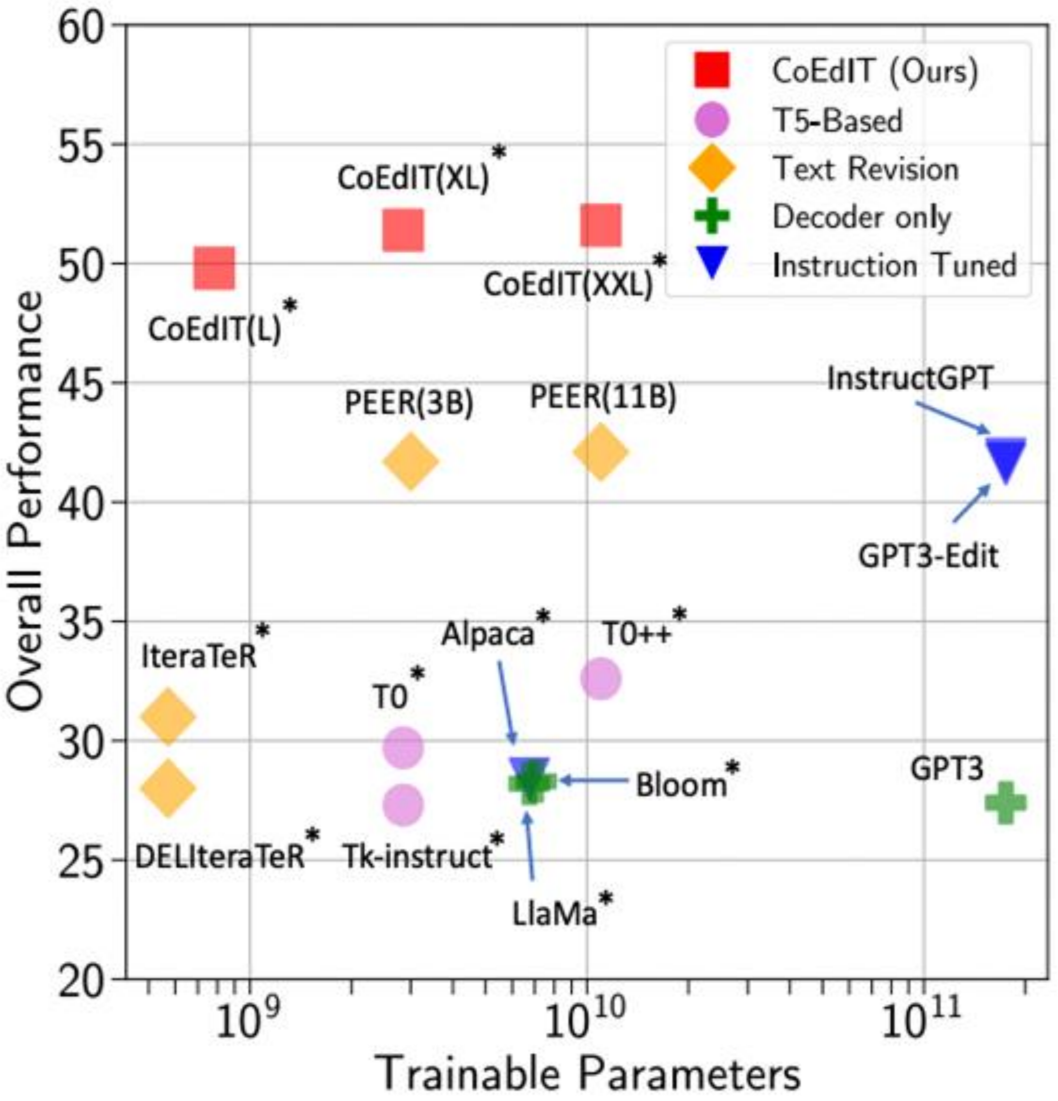


| Edit Intention | Datasets | Size | Example Input | Example Output |
|---|---|---|---|---|
| FLUENCY | NUCLE-14<br>Lang-8<br>BEA-19 | 20*k* | *Fix the grammar:* When I grow up, I start to understand what he said is quite right. | When I grew up, I started to understand what he said was quite right. |
| COHERENCE | DiscoFuse | 11*k* | *Make this text coherent:* Their flight is weak. They run quickly through the tree canopy. | Their flight is weak, but they run quickly through the tree canopy. |
| CLARITY (Simplification) | NEWSELA<br>WikiLarge<br>WikiAuto<br>ParabankV2<br>ITERATER-CLARITY | 13*k* | *Rewrite to make this easier to understand:* A storm surge is what forecasters consider a hurricane's most treacherous aspect. | A storm surge is considered a hurricane's most dangerous aspect. |
| STYLE (Paraphrase) | ParabankV2 | 15*k* | *Paraphrase this:* Do you know where I was born? | Do you know my birthplace? |
| STYLE (Formalize) | GYAFC | 12*k* | *Write this more formally:* omg i love that song im listening to it right now | I love that song and I am listening to it at this moment. |
| STYLE (Neutralize) | WNC | 11*k* | *Write in a more neutral way:* The authors' exposé on nutrition studies. | The authors' statements on nutrition studies. |

Table 1: Example data instances in the CoEdIT dataset (82K `<instruction: source, target>` pairs). Instructional prompts in the inputs are *italicized*.

[Raheja et al., EMNLP Findings 23]

# Quantitative Evaluations of Text Revision Models

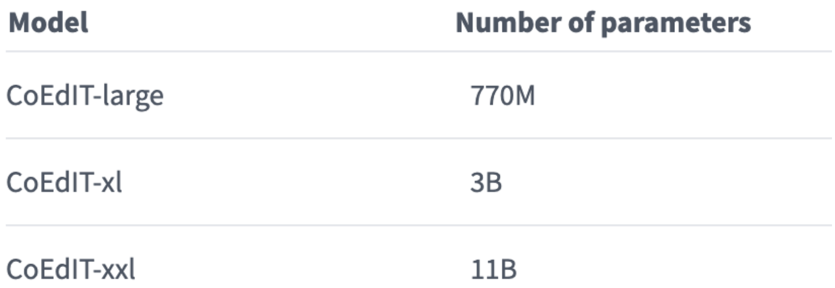


CoEdIT's task specific instruction tuning largely outperforms most generally instruction-tuned models

Overall, CoEdIT generates better text edits than models that are even 60x larger

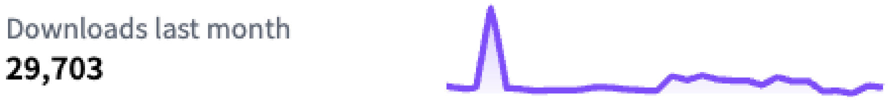
[Raheja et al., EMNLP Findings 23]

# CoEdit on HuggingFace



| Model | Number of parameters |
|---|---|
| CoEdIT-large | 770M |
| CoEdIT-xl | 3B |
| CoEdIT-xxl | 11B |

[https://huggingface.co/grammarly/coedit-large](https://huggingface.co/grammarly/coedit-large)



Downloads last month

**29,703**

```
from transformers import AutoTokenizer, T5ForConditionalGeneration

tokenizer = AutoTokenizer.from_pretrained("grammarly/coedit-large")
model = T5ForConditionalGeneration.from_pretrained("grammarly/coedit-large")
input_text = 'Fix grammatical errors in this sentence: When I grow up, I start to understand what he said is quite right.'
input_ids = tokenizer(input_text, return_tensors="pt").input_ids
outputs = model.generate(input_ids, max_length=256)
edited_text = tokenizer.decode(outputs[0], skip_special_tokens=True)
```

[Raheja et al., EMNLP Findings 23]

# Augmented Language Models:

# From Base to Instruct/Chat Models

| Going from base models… | …to instruct/chat models |
|---|---|
| Llama3.2-1B | Llama3.2-1B-Instruct |
| Llama3.2-3B | Llama3.2-3B-Instruct |
| Qwen2.5-0.5B | Qwen2.5-0.5B-Instruct |
| Falcon3-1B-Base | Falcon3-1B-Instruct |
| … | … |

Pretraining

Post-training (of which part is Instruction-Tuning)

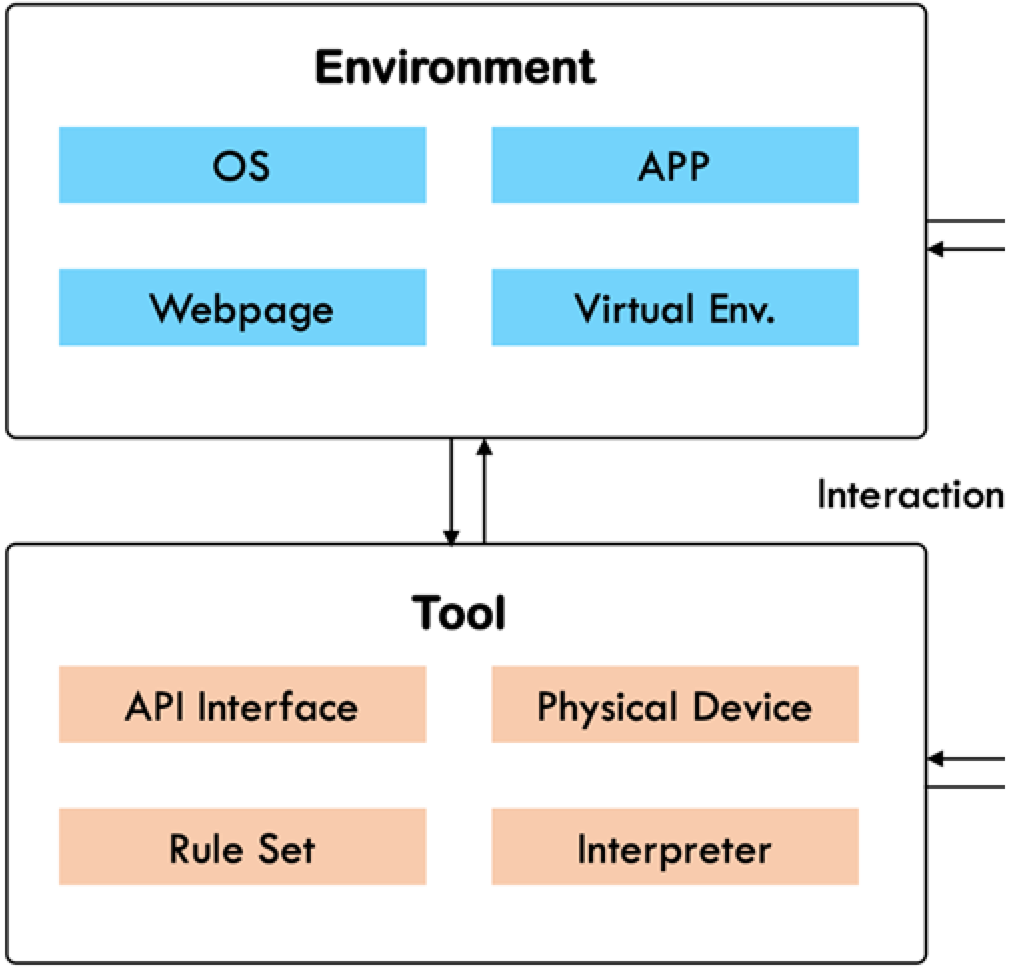
# Weakness of LLMs

- Pre-trained LLMs struggle at completing tasks that require:
  - Latest knowledge after the model pretraining time cutoff or
  - Knowledge with internal/private knowledge base
  - Symbolic or other deterministic execution capabilities
- These issues stem from their fundamental limitations:
  - They are trained to perform statistical modeling given a single parametric model and a limited context
  - Their main objective function, *the next token prediction task*, does not cater for explicit symbolic capabilities

# Augmented Language Models

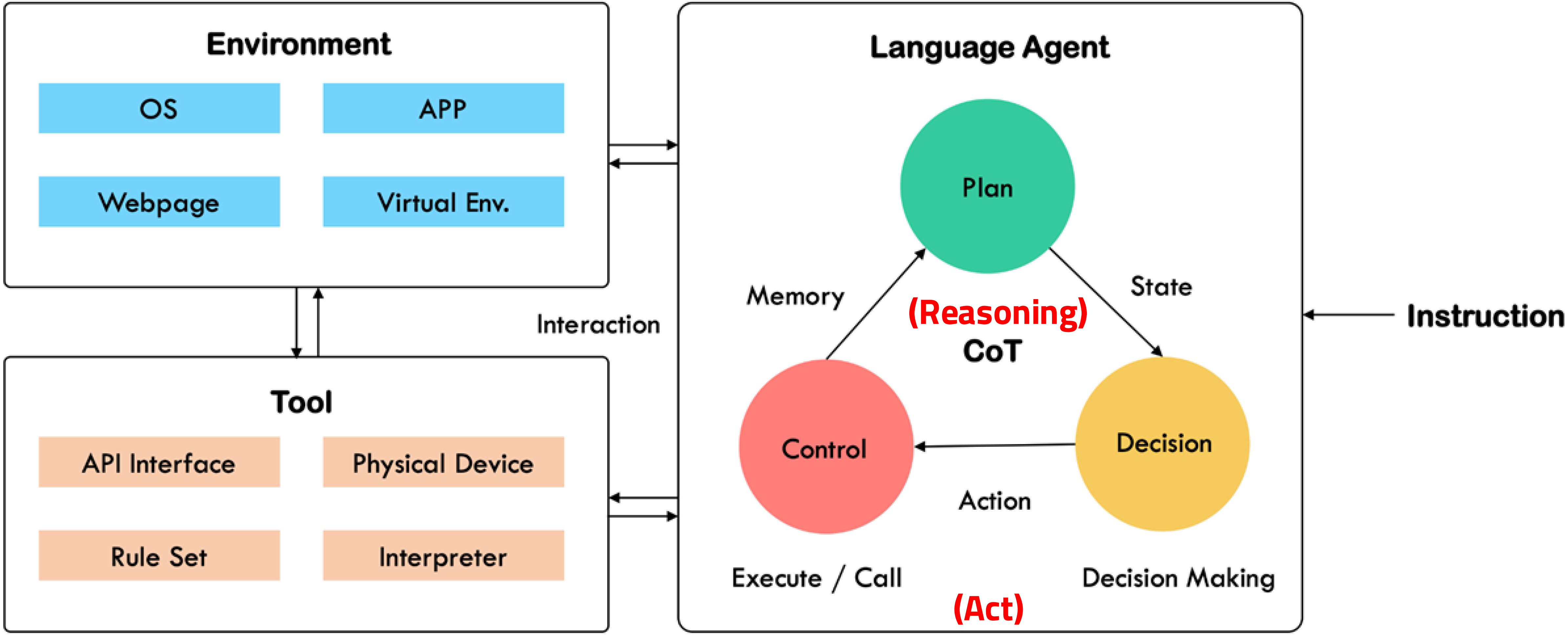
- Recent trend is to move slightly away from the purely statistical language modeling and *integrate external components*
  - So that a *more relevant context* is produced at the cost of more computation
  - Resulting in non-parametric models
- An augmentation can be viewed in three dimensions: [Mialon et al. 2023]
  - **Reasoning**: breaking up a complex task into smaller subtasks
  - **Tool**: external modules that can be called
  - **Act**: Calling of a tool to have an effect

# Augmented Language Models



[Zhang et al. 2023]

# Augmented Language Models



[Zhang et al. 2023]

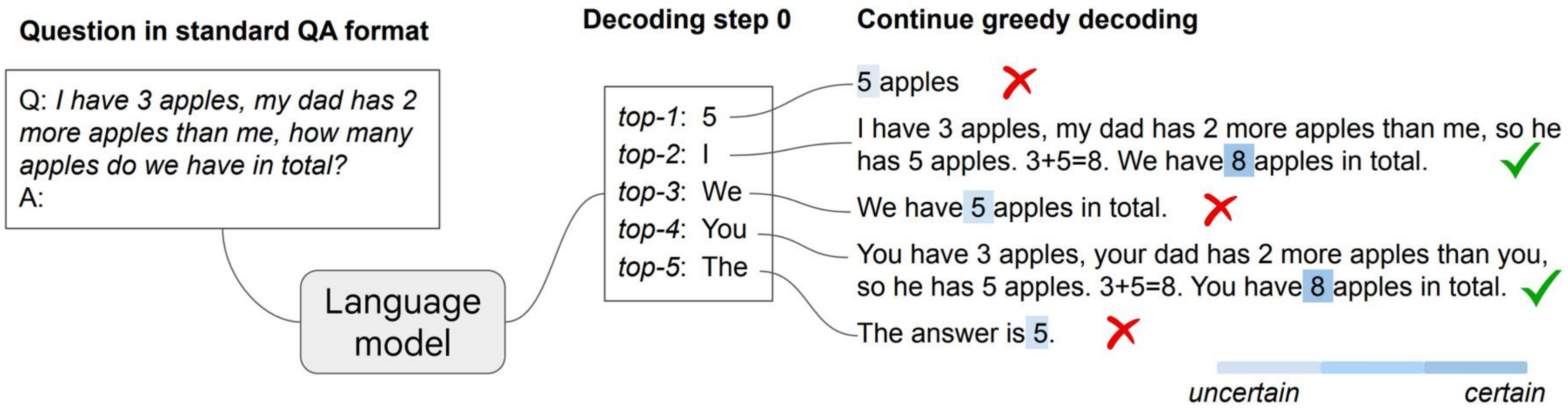
# Reasoning

# Reasoning

- Reasoning is the ability to make inferences using evidence and logic.
  - Commonsense, mathematical, symbolic, etc.
  - Often this involves deductions from inference chains, i.e. “multi-step reasoning”
- Main challenge is to break down a complex problem into smaller subproblems and generate the solution by composing the (correctly predicted) answers to the subproblems.
- Eliciting reasoning in LLMs
  - Eliciting reasoning with prompting
  - Divide and concur with recursive prompting
  - Teaching LLMs to reason

# Eliciting reason with prompting

- Essentially methods can be categorized as either zero-shot or few-shot
- Zero-shot prompting
  - "Let's think step by step." [Kojima et al. 2022]
  - "Chain-of-thought decoding" [Wang and Zhou 2024]



Q: *I have 3 apples, my dad has 2 more apples than me, how many apples do we have in total?*
A:

5 apples

I have 3 apples, my dad has 2 more apples than me, so he has 5 apples. 3+5=8. We have 8 apples in total.

We have 5 apples in total.

You have 3 apples, your dad has 2 more apples than you, so he has 5 apples. 3+5=8. You have 8 apples in total.

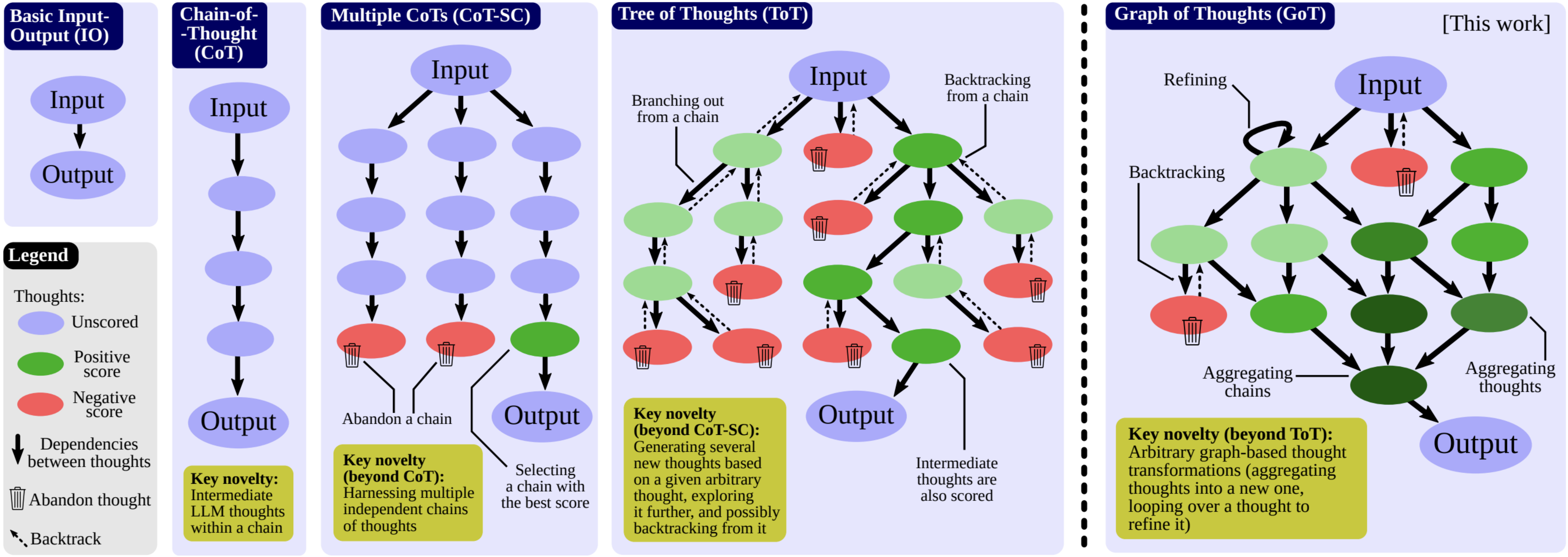
The answer is 5.

Figure 1 | **Illustration of CoT-decoding.** Pre-trained LLMs are capable of inherent reasoning without prompting by considering alternative top-$k$ tokens, rather than solely relying on the top-1 greedy decoding path. Moreover, these models tend to display higher confidence in decoding the final answer (indicated by a darker shaded color) when a CoT reasoning path is present.

# Eliciting reason with prompting

- Few-shot prompting – variants of CoT prompting

[Besta et al. 2024]



**Key novelty:** Intermediate LLM thoughts within a chain

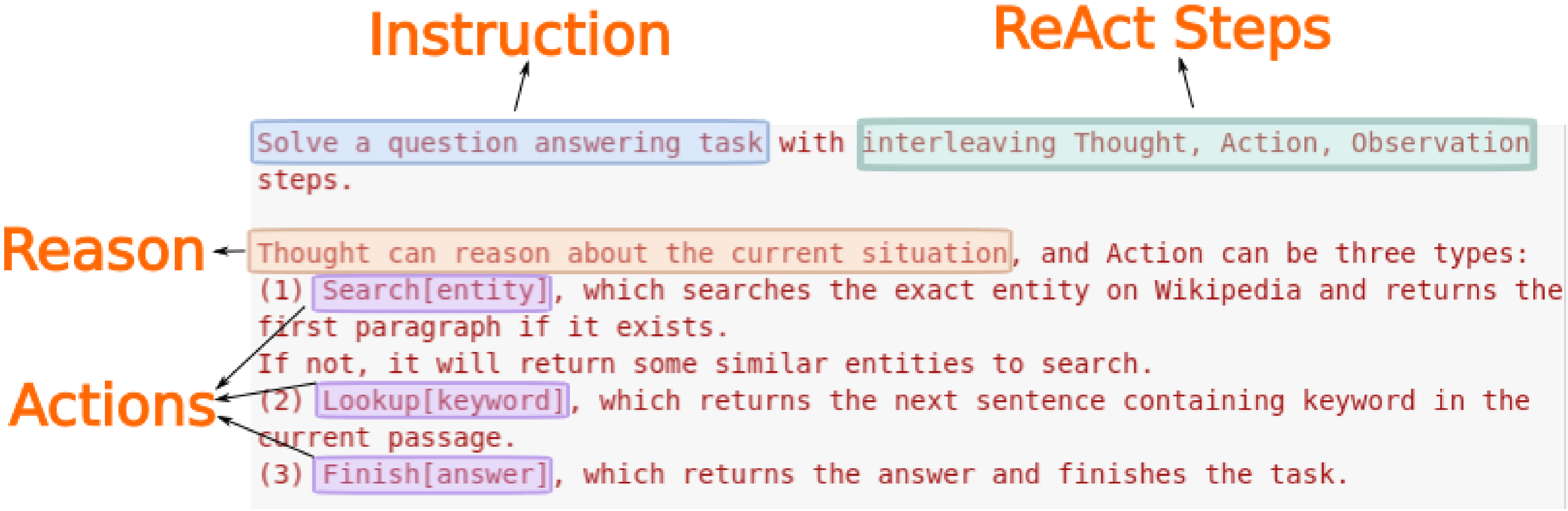
**Key novelty (beyond CoT):** Harnessing multiple independent chains of thoughts

**Key novelty (beyond CoT-SC):** Generating several new thoughts based on a given arbitrary thought, exploring it further, and possibly backtracking from it

**Key novelty (beyond ToT):** Arbitrary graph-based thought transformations (aggregating thoughts into a new one, looping over a thought to refine it)

# Eliciting reason with prompting

- Few-shot prompting – via "programming"
  - ReAct prompting [Yao et al. 2022]



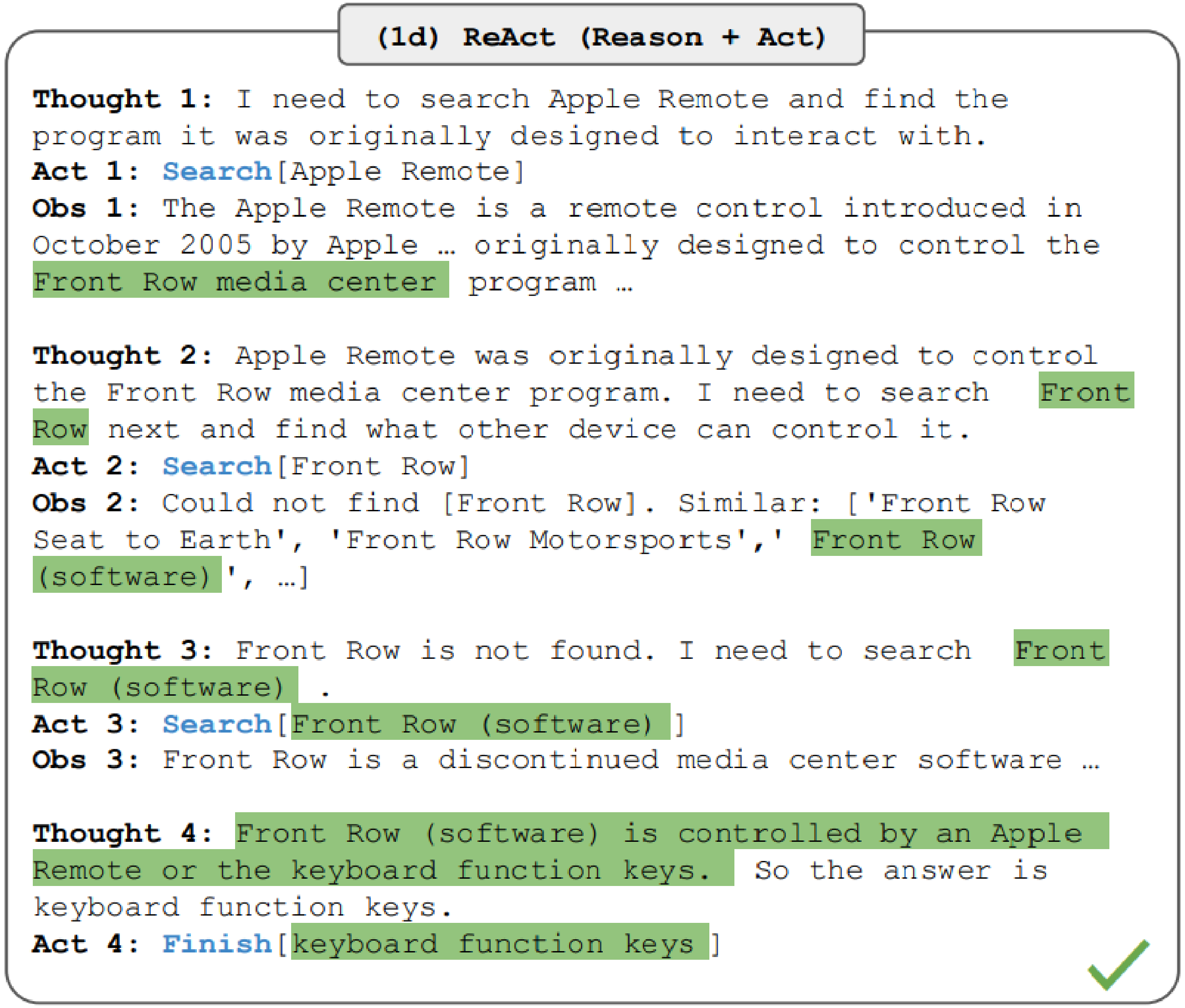
Solve a question answering task with interleaving Thought, Action, Observation steps.

Thought can reason about the current situation, and Action can be three types:
(1) Search[entity], which searches the exact entity on Wikipedia and returns the first paragraph if it exists.
If not, it will return some similar entities to search.
(2) Lookup[keyword], which returns the next sentence containing keyword in the current passage.
(3) Finish[answer], which returns the answer and finishes the task.

https://www.width.ai/post/react-prompting

# Eliciting reason with prompting

- ❑ Few-shot prompting – via "programming"
    - ○ ReAct prompting [Yao et al. 2022]



**(1d) ReAct (Reason + Act)**

**Thought 1:** I need to search Apple Remote and find the program it was originally designed to interact with.
**Act 1:** **Search**[Apple Remote]
**Obs 1:** The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

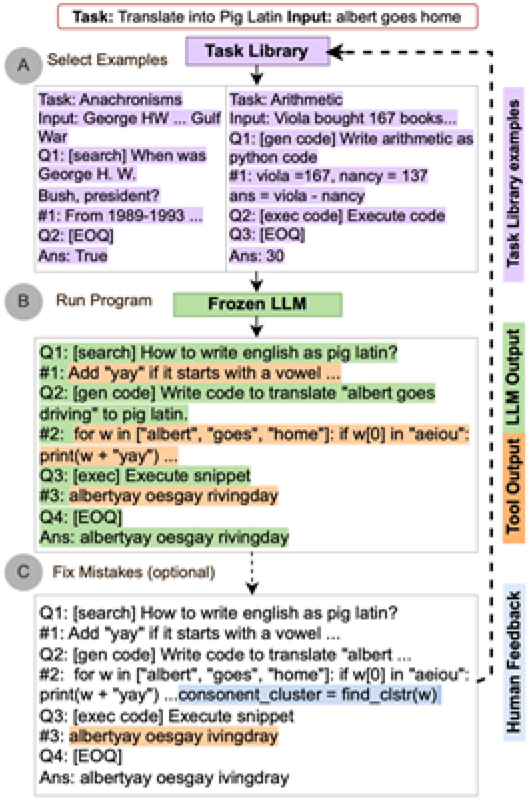
**Thought 2:** Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
**Act 2:** **Search**[Front Row]
**Obs 2:** Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

**Thought 3:** Front Row is not found. I need to search Front Row (software) .
**Act 3:** **Search**[Front Row (software) ]
**Obs 3:** Front Row is a discontinued media center software …

**Thought 4:** Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
**Act 4:** **Finish**[keyboard function keys ]

✓

# Programing LLMs for *multi-step reasoning*



```
Task: Anachronisms
Input: George HW ... Gulf War
Q1: [search] When was George H. W. Bush, president?
#1: From 1989-1993 ...
Q2: [EOQ]
Ans: True
```

```
Task: Arithmetic
Input: Viola bought 167 books...
Q1: [gen code] Write arithmetic as python code
#1: viola =167, nancy = 137
ans = viola - nancy
Q2: [exec code] Execute code
Q3: [EOQ]
Ans: 30
```

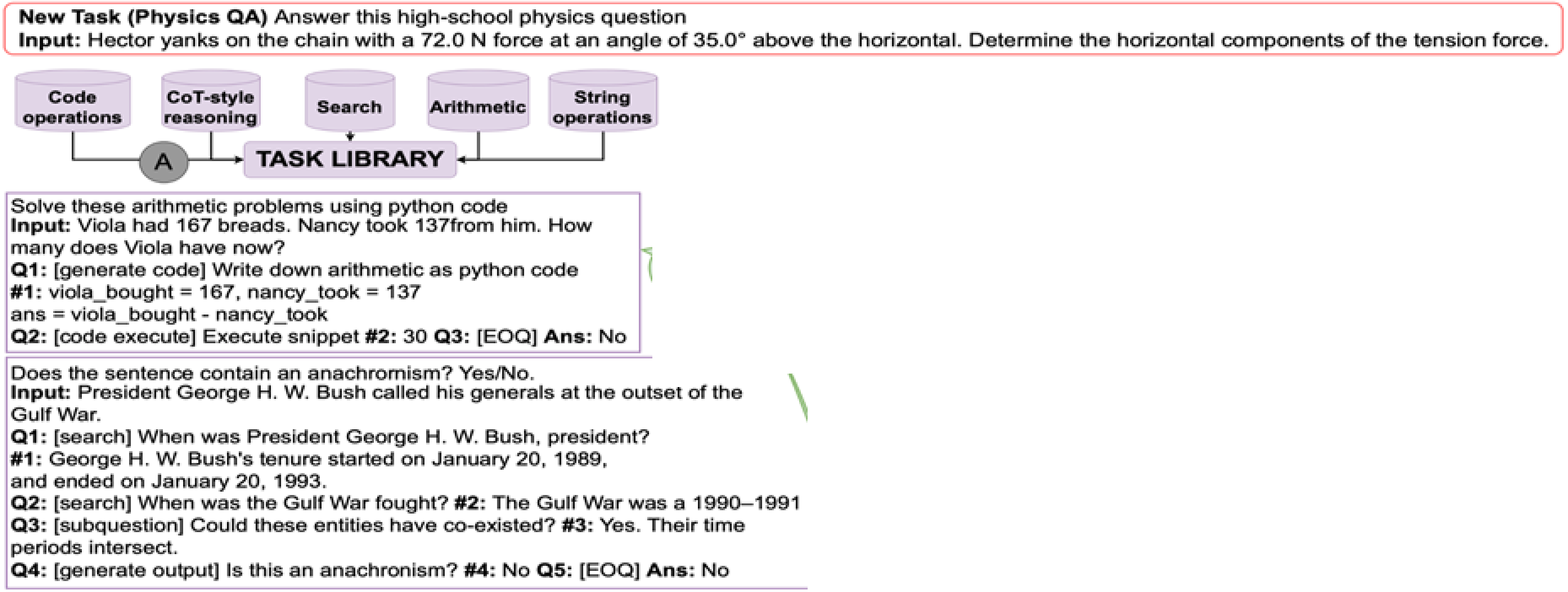
```
Q1: [search] How to write english as pig latin?
#1: Add "yay" if it starts with a vowel ...
Q2: [gen code] Write code to translate "albert goes driving" to pig latin.
#2:  for w in ["albert", "goes", "home"]: if w[0] in "aeiou":
print(w + "yay") ...
Q3: [exec] Execute snippet
#3: albertyay oesgay rivingday
Q4: [EOQ]
Ans: albertyay oesgay rivingday
```

```
Q1: [search] How to write english as pig latin?
#1: Add "yay" if it starts with a vowel ...
Q2: [gen code] Write code to translate "albert ...
#2:  for w in ["albert", "goes", "home"]: if w[0] in "aeiou":
print(w + "yay") ...consonent_cluster = find_clstr(w)
Q3: [exec code] Execute snippet
#3: albertyay oesgay ivingdray
Q4: [EOQ]
Ans: albertyay oesgay ivingdray
```

Figure 1: ART generates automatic multi-step decompositions for new tasks by selecting decompositions of related tasks in the *task library* (A) and selecting and using tools in the *tool library* alongside LLM generation (B). Humans can optionally edit decompositions (eg. correcting and editing code) to improve performance (C).

- ❑ **ART** (Automatic multi-step reasoning and tool-use for large language models; [Paranjape et al. 2023](#)):
  - ART automatically creates decompositions (multistep reasoning) for examples of new tasks.
  - ART retrieves comparable task instances from a library, enabling quick task analysis and tool application.
  - Using a structured query language, it facilitates reading intermediate stages, pausing for external tool use, and restarting after tool output integration.
    - ✓ At each step, the framework selects and utilizes the most appropriate tools.

# Programing LLMs for *multi-step reasoning*



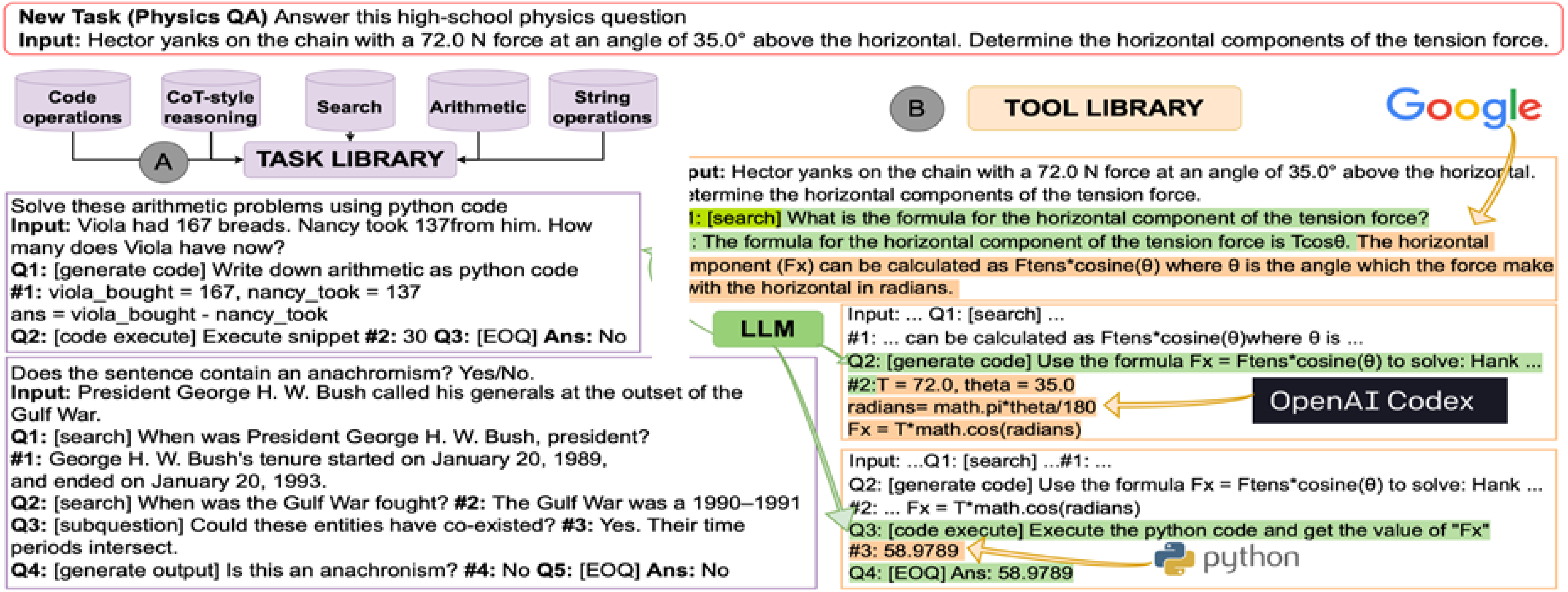
**New Task (Physics QA)** Answer this high-school physics question
**Input:** Hector yanks on the chain with a 72.0 N force at an angle of 35.0° above the horizontal. Determine the horizontal components of the tension force.

Solve these arithmetic problems using python code
**Input:** Viola had 167 breads. Nancy took 137from him. How many does Viola have now?
**Q1:** [generate code] Write down arithmetic as python code
**#1:** viola_bought = 167, nancy_took = 137
ans = viola_bought - nancy_took
**Q2:** [code execute] Execute snippet **#2:** 30 **Q3:** [EOQ] **Ans:** No

Does the sentence contain an anachromism? Yes/No.
**Input:** President George H. W. Bush called his generals at the outset of the Gulf War.
**Q1:** [search] When was President George H. W. Bush, president?
**#1:** George H. W. Bush's tenure started on January 20, 1989, and ended on January 20, 1993.
**Q2:** [search] When was the Gulf War fought? **#2:** The Gulf War was a 1990–1991
**Q3:** [subquestion] Could these entities have co-existed? **#3:** Yes. Their time periods intersect.
**Q4:** [generate output] Is this an anachronism? **#4:** No **Q5:** [EOQ] **Ans:** No

Figure 2: A run-through of ART on a new task, Physics QA. (A) Programs of related tasks like anachronisms and Math QA provide few-shot supervision to the LLM — related sub-steps and tools in these programs can be used by the LLM for cross-task generalization (shown in purple). (B) Tool use: Search is used to find the appropriate physics formula, and code generation and execution are used to substitute given values and compute the answer (shown in orange).

# Programing LLMs for *multi-step reasoning*



**New Task (Physics QA)** Answer this high-school physics question
**Input:** Hector yanks on the chain with a 72.0 N force at an angle of 35.0° above the horizontal. Determine the horizontal components of the tension force.

Solve these arithmetic problems using python code
**Input:** Viola had 167 breads. Nancy took 137from him. How many does Viola have now?
**Q1:** [generate code] Write down arithmetic as python code
**#1:** viola_bought = 167, nancy_took = 137
ans = viola_bought - nancy_took
**Q2:** [code execute] Execute snippet **#2:** 30 **Q3:** [EOQ] **Ans:** No

Does the sentence contain an anachromism? Yes/No.
**Input:** President George H. W. Bush called his generals at the outset of the Gulf War.
**Q1:** [search] When was President George H. W. Bush, president?
**#1:** George H. W. Bush's tenure started on January 20, 1989, and ended on January 20, 1993.
**Q2:** [search] When was the Gulf War fought? **#2:** The Gulf War was a 1990–1991
**Q3:** [subquestion] Could these entities have co-existed? **#3:** Yes. Their time periods intersect.
**Q4:** [generate output] Is this an anachronism? **#4:** No **Q5:** [EOQ] **Ans:** No

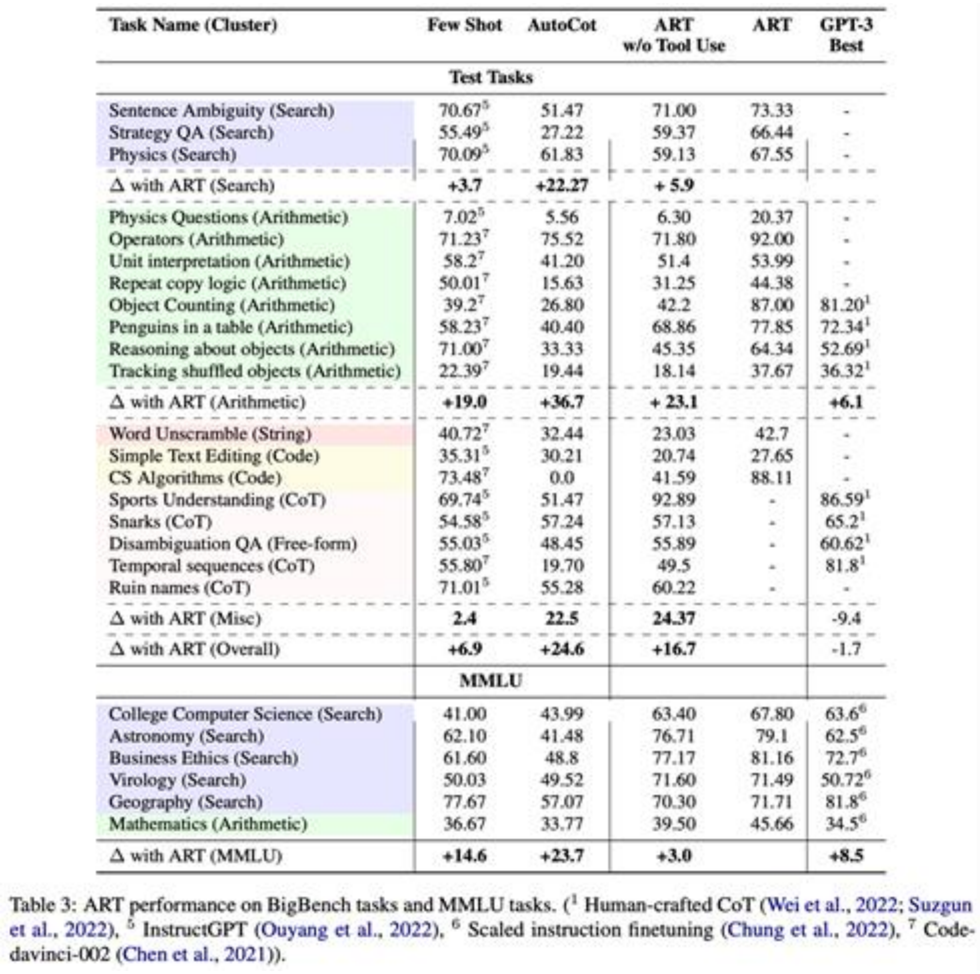
**put:** Hector yanks on the chain with a 72.0 N force at an angle of 35.0° above the horizontal.
etermine the horizontal components of the tension force.
**1:** [search] What is the formula for the horizontal component of the tension force?
: The formula for the horizontal component of the tension force is Tcosθ. The horizontal
mponent (Fx) can be calculated as Ftens*cosine(θ) where θ is the angle which the force make
with the horizontal in radians.

Input: ... Q1: [search] ...
#1: ... can be calculated as Ftens*cosine(θ)where θ is ...
Q2: [generate code] Use the formula Fx = Ftens*cosine(θ) to solve: Hank ...
#2:T = 72.0, theta = 35.0
radians= math.pi*theta/180
Fx = T*math.cos(radians)

Input: ...Q1: [search] ...#1: ...
Q2: [generate code] Use the formula Fx = Ftens*cosine(θ) to solve: Hank ...
#2: ... Fx = T*math.cos(radians)
Q3: [code execute] Execute the python code and get the value of "Fx"
#3: 58.9789
Q4: [EOQ] Ans: 58.9789

Figure 2: A run-through of ART on a new task, Physics QA. (A) Programs of related tasks like anachronisms and Math QA provide few-shot supervision to the LLM — related sub-steps and tools in these programs can be used by the LLM for cross-task generalization (shown in purple). (B) Tool use: Search is used to find the appropriate physics formula, and code generation and execution are used to substitute given values and compute the answer (shown in orange).

# Programing LLMs for *multi-step reasoning*



| Task Name (Cluster) | Few Shot | AutoCot | ART w/o Tool Use | ART | GPT-3 Best |
|---|---|---|---|---|---|
| | **Test Tasks** | | | | |
| Sentence Ambiguity (Search) | 70.67[5] | 51.47 | 71.00 | 73.33 | - |
| Strategy QA (Search) | 55.49[5] | 27.22 | 59.37 | 66.44 | - |
| Physics (Search) | 70.09[5] | 61.83 | 59.13 | 67.55 | - |
| Δ with ART (Search) | **+3.7** | **+22.27** | **+ 5.9** | | |
| Physics Questions (Arithmetic) | 7.02[5] | 5.56 | 6.30 | 20.37 | - |
| Operators (Arithmetic) | 71.23[7] | 75.52 | 71.80 | 92.00 | - |
| Unit interpretation (Arithmetic) | 58.2[7] | 41.20 | 51.4 | 53.99 | - |
| Repeat copy logic (Arithmetic) | 50.01[7] | 15.63 | 31.25 | 44.38 | - |
| Object Counting (Arithmetic) | 39.2[7] | 26.80 | 42.2 | 87.00 | 81.20[1] |
| Penguins in a table (Arithmetic) | 58.23[7] | 40.40 | 68.86 | 77.85 | 72.34[1] |
| Reasoning about objects (Arithmetic) | 71.00[7] | 33.33 | 45.35 | 64.34 | 52.69[1] |
| Tracking shuffled objects (Arithmetic) | 22.39[7] | 19.44 | 18.14 | 37.67 | 36.32[1] |
| Δ with ART (Arithmetic) | **+19.0** | **+36.7** | **+ 23.1** | | **+6.1** |
| Word Unscramble (String) | 40.72[7] | 32.44 | 23.03 | 42.7 | - |
| Simple Text Editing (Code) | 35.31[5] | 30.21 | 20.74 | 27.65 | - |
| CS Algorithms (Code) | 73.48[7] | 0.0 | 41.59 | 88.11 | - |
| Sports Understanding (CoT) | 69.74[5] | 51.47 | 92.89 | - | 86.59[1] |
| Snarks (CoT) | 54.58[5] | 57.24 | 57.13 | - | 65.2[1] |
| Disambiguation QA (Free-form) | 55.03[5] | 48.45 | 55.89 | - | 60.62[1] |
| Temporal sequences (CoT) | 55.80[7] | 19.70 | 49.5 | - | 81.8[1] |
| Ruin names (CoT) | 71.01[5] | 55.28 | 60.22 | - | - |
| Δ with ART (Misc) | **2.4** | **22.5** | **24.37** | | -9.4 |
| Δ with ART (Overall) | **+6.9** | **+24.6** | **+16.7** | | -1.7 |
| | **MMLU** | | | | |
| College Computer Science (Search) | 41.00 | 43.99 | 63.40 | 67.80 | 63.6[6] |
| Astronomy (Search) | 62.10 | 41.48 | 76.71 | 79.1 | 62.5[6] |
| Business Ethics (Search) | 61.60 | 48.8 | 77.17 | 81.16 | 72.7[6] |
| Virology (Search) | 50.03 | 49.52 | 71.60 | 71.49 | 50.72[6] |
| Geography (Search) | 77.67 | 57.07 | 70.30 | 71.71 | 81.8[6] |
| Mathematics (Arithmetic) | 36.67 | 33.77 | 39.50 | 45.66 | 34.5[6] |
| Δ with ART (MMLU) | **+14.6** | **+23.7** | **+3.0** | | **+8.5** |

Table 3: ART performance on BigBench tasks and MMLU tasks. ([1] Human-crafted CoT (Wei et al., 2022; Suzgun et al., 2022), [5] InstructGPT (Ouyang et al., 2022), [6] Scaled instruction finetuning (Chung et al., 2022), [7] Code-davinci-002 (Chen et al., 2021)).

# Using Tools and Act

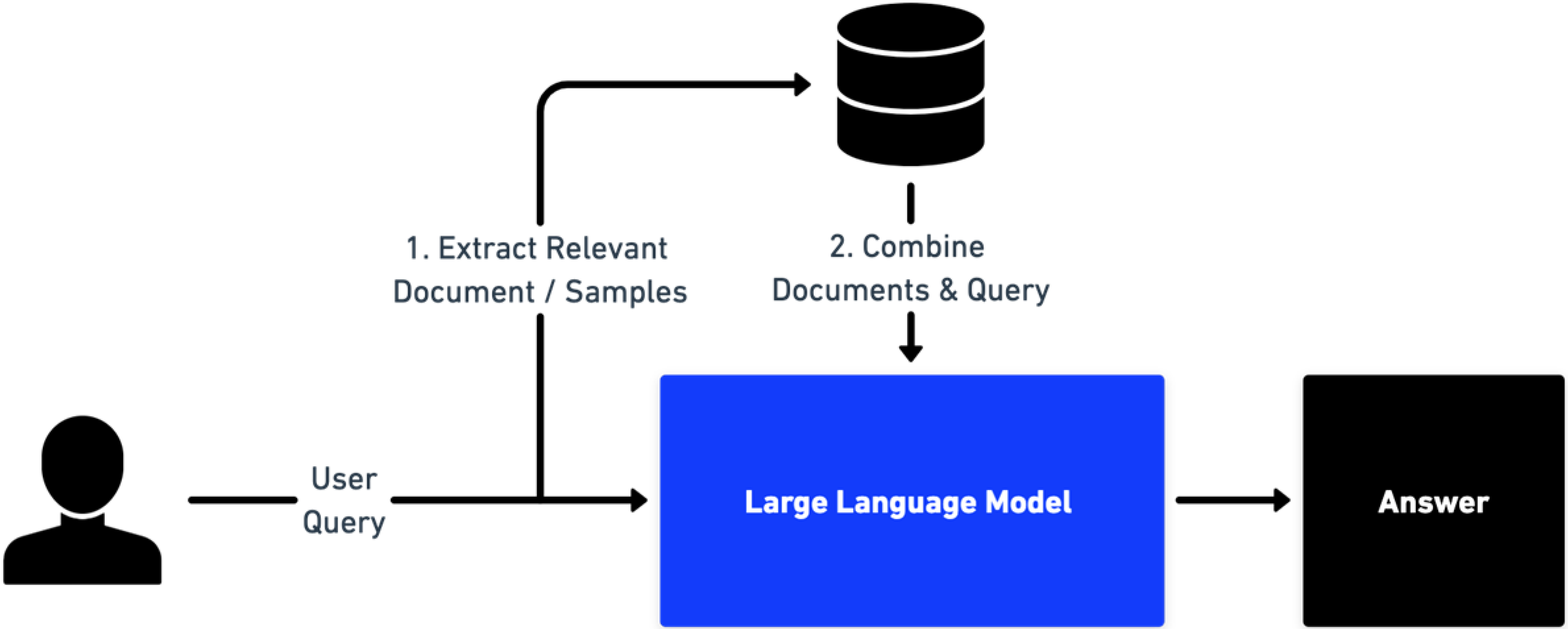
# Tools and Act

- Expanding LLMs' capabilities to access and leverage external knowledge and computational resources beyond what is stored in their own weights.
- This integration, via special tokens, allows LLMs to perform tasks like:
  - Exact computation
  - Symbolic reasoning
  - Information retrieval, etc.
- Tools can range from:
  - Calling (another) LLM or other programs like Python interpreter
  - Querying (vector) databases, knowledge graphs, or search engines
    - Often referred to as "information retrieval"

# Tools and Act

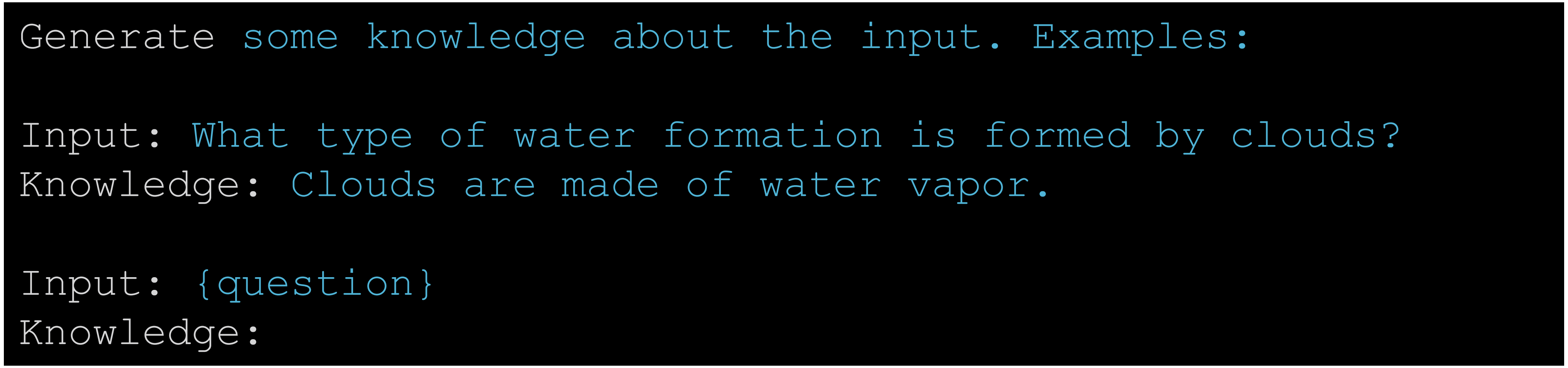
- Use Google Search for document retrieval to augment LLMs. [Lazaridou et al. 2022]
  - Given a question, clean text (paragraph) is extracted out of 20 URLs returned by Google.
  - Paragraphs are ranked by TF-IDF based cosine similarity between evidence paragraphs and the query.
  - Only the most relevant paragraph is used in the prompt to produce an answer

# Augmenting LLMs with Retrieval


1. Extract Relevant Document / Samples
2. Combine Documents & Query
User Query
Large Language Model
Answer

# Augmenting LLMs with Retrieval

- ❑ "Internal retrieval" from LLMs could be also beneficial
  - ○ Generate knowledge about a topic before answering the question [Liu et al. 2022]



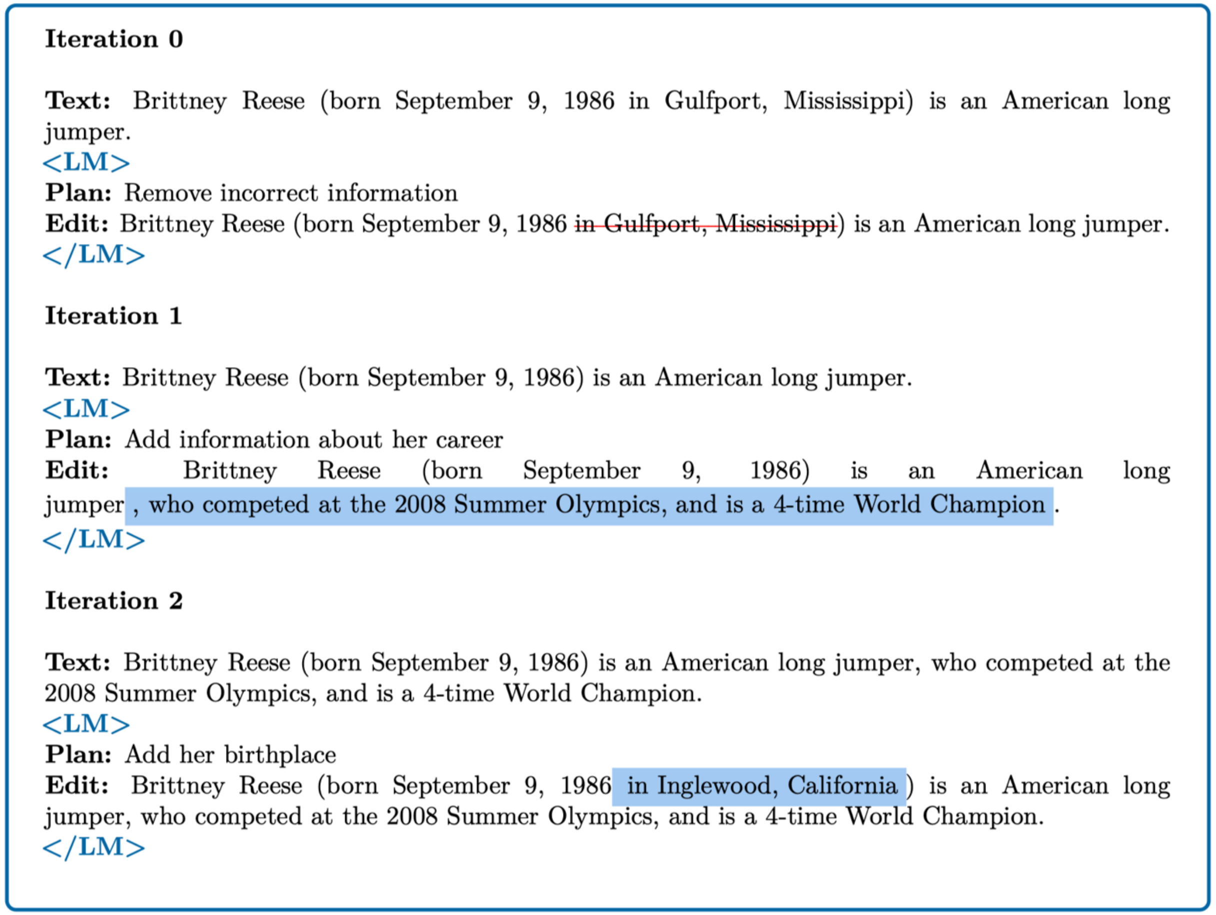
```
Generate some knowledge about the input. Examples:

Input: What type of water formation is formed by clouds?
Knowledge: Clouds are made of water vapor.

Input: {question}
Knowledge:
```

# Augmenting LLMs with Retrieval

- “Internal retrieval” from LLMs could be also beneficial
  - Iterative prompting of LLM for text refinement task [Schick et al. 2022]
  - The “plan” can come from human users too.



**Iteration 0**

**Text:** Brittney Reese (born September 9, 1986 in Gulfport, Mississippi) is an American long jumper.
<LM>
**Plan:** Remove incorrect information
**Edit:** Brittney Reese (born September 9, 1986 ~~in Gulfport, Mississippi~~) is an American long jumper.
</LM>

**Iteration 1**

**Text:** Brittney Reese (born September 9, 1986) is an American long jumper.
<LM>
**Plan:** Add information about her career
**Edit:** Brittney Reese (born September 9, 1986) is an American long jumper , who competed at the 2008 Summer Olympics, and is a 4-time World Champion .
</LM>

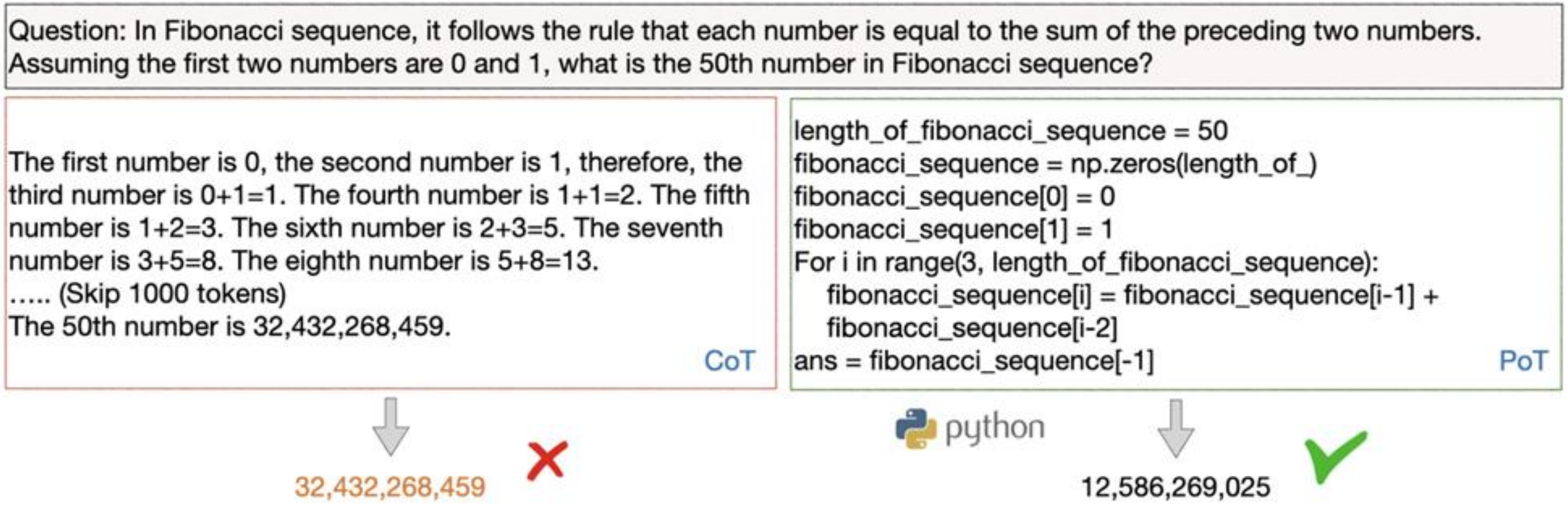
**Iteration 2**

**Text:** Brittney Reese (born September 9, 1986) is an American long jumper, who competed at the 2008 Summer Olympics, and is a 4-time World Champion.
<LM>
**Plan:** Add her birthplace
**Edit:** Brittney Reese (born September 9, 1986 in Inglewood, California ) is an American long jumper, who competed at the 2008 Summer Olympics, and is a 4-time World Champion.
</LM>

Figure 5: Iterative prompting example using PEER (Schick et al., 2022), a LM trained to produce a plan of action and edit to the input text at each step. This process can be repeated until the generated text requires no further updates. <LM> denotes the start of the LM’s output to the prompt, while </LM> denotes the end.

# Augmenting LLMs with Programming Language

- Ask LLM to generate programming language statements to resolve natural language reasoning problems, hence offloading the solution step to a runtime such as a Python interpreter.



Question: In Fibonacci sequence, it follows the rule that each number is equal to the sum of the preceding two numbers. Assuming the first two numbers are 0 and 1, what is the 50th number in Fibonacci sequence?

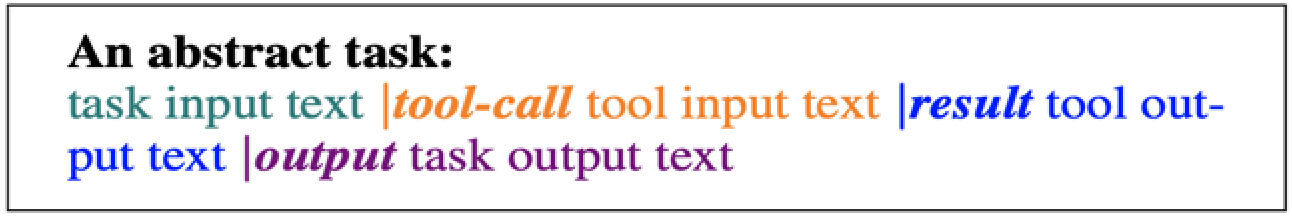
The first number is 0, the second number is 1, therefore, the third number is 0+1=1. The fourth number is 1+1=2. The fifth number is 1+2=3. The sixth number is 2+3=5. The seventh number is 3+5=8. The eighth number is 5+8=13.
..... (Skip 1000 tokens)
The 50th number is 32,432,268,459.

```
length_of_fibonacci_sequence = 50
fibonacci_sequence = np.zeros(length_of_)
fibonacci_sequence[0] = 0
fibonacci_sequence[1] = 1
For i in range(3, length_of_fibonacci_sequence):
   fibonacci_sequence[i] = fibonacci_sequence[i-1] +
   fibonacci_sequence[i-2]
ans = fibonacci_sequence[-1]
```

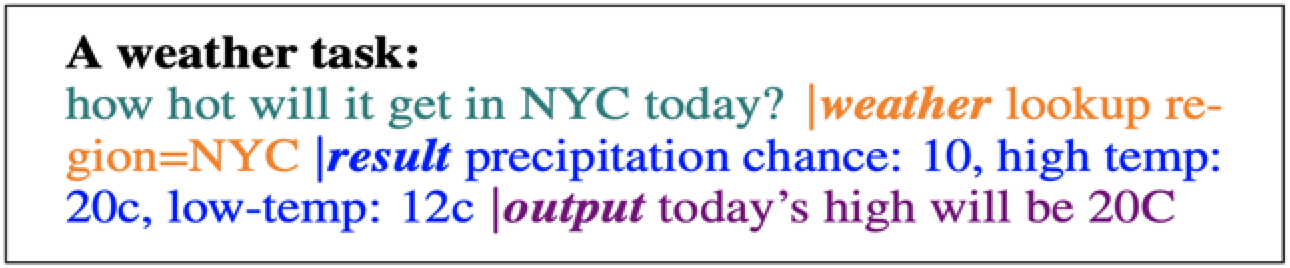
**PAL** (Program-aided language models); Gao et al. 2022 and **PoT** (Program of Thoughts prompting); Chen et al. 2022

# Augmenting LLMs with External APIs

- **TALM** (Tool Augmented Language Models; [Parisi et al. 2022]): LM augmented with text-to-text API calls.
  - LM is guided to generate |tool-call and tool input text conditioned on task input text to construct API call requests
  - When |result shows up, the specified tool API is called and the returned result gets appended to the text sequence as |output token.



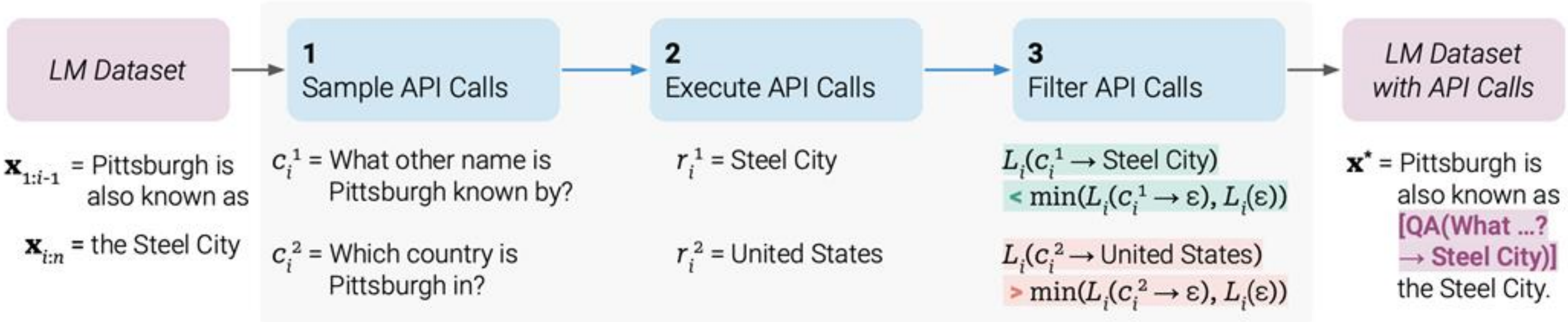
**An abstract task:**
task input text |***tool-call*** tool input text |***result*** tool output text |***output*** task output text



**A weather task:**
how hot will it get in NYC today? |***weather*** lookup region=NYC |***result*** precipitation chance: 10, high temp: 20c, low-temp: 12c |***output*** today's high will be 20C

# Augmenting LLMs with External APIs

- **Toolformer** (Schick et al. 2023) use external tools via simple APIs, which is built in a *self-supervised manner* and only requires a handful of demonstrations for each API.
  - **Calculator** to help LM with the lack of precise math skills;
  - **Q&A** system to help with unfaithful content and hallucination;
  - **Search engine** to provide up-to-date information after pretraining cut off time;
  - **Translation system** to improve performance on low resource language;
  - **Calendar** to make LM be aware of time progression.



*LM Dataset* → **1** Sample API Calls → **2** Execute API Calls → **3** Filter API Calls → *LM Dataset with API Calls*

$\mathbf{x}_{1:i-1}$ = Pittsburgh is also known as

$\mathbf{x}_{i:n}$ = the Steel City

$c_i^1$ = What other name is Pittsburgh known by?

$c_i^2$ = Which country is Pittsburgh in?

$r_i^1$ = Steel City

$r_i^2$ = United States

$L_i(c_i^1 \rightarrow \text{Steel City}) < \min(L_i(c_i^1 \rightarrow \varepsilon), L_i(\varepsilon))$

$L_i(c_i^2 \rightarrow \text{United States}) > \min(L_i(c_i^2 \rightarrow \varepsilon), L_i(\varepsilon))$

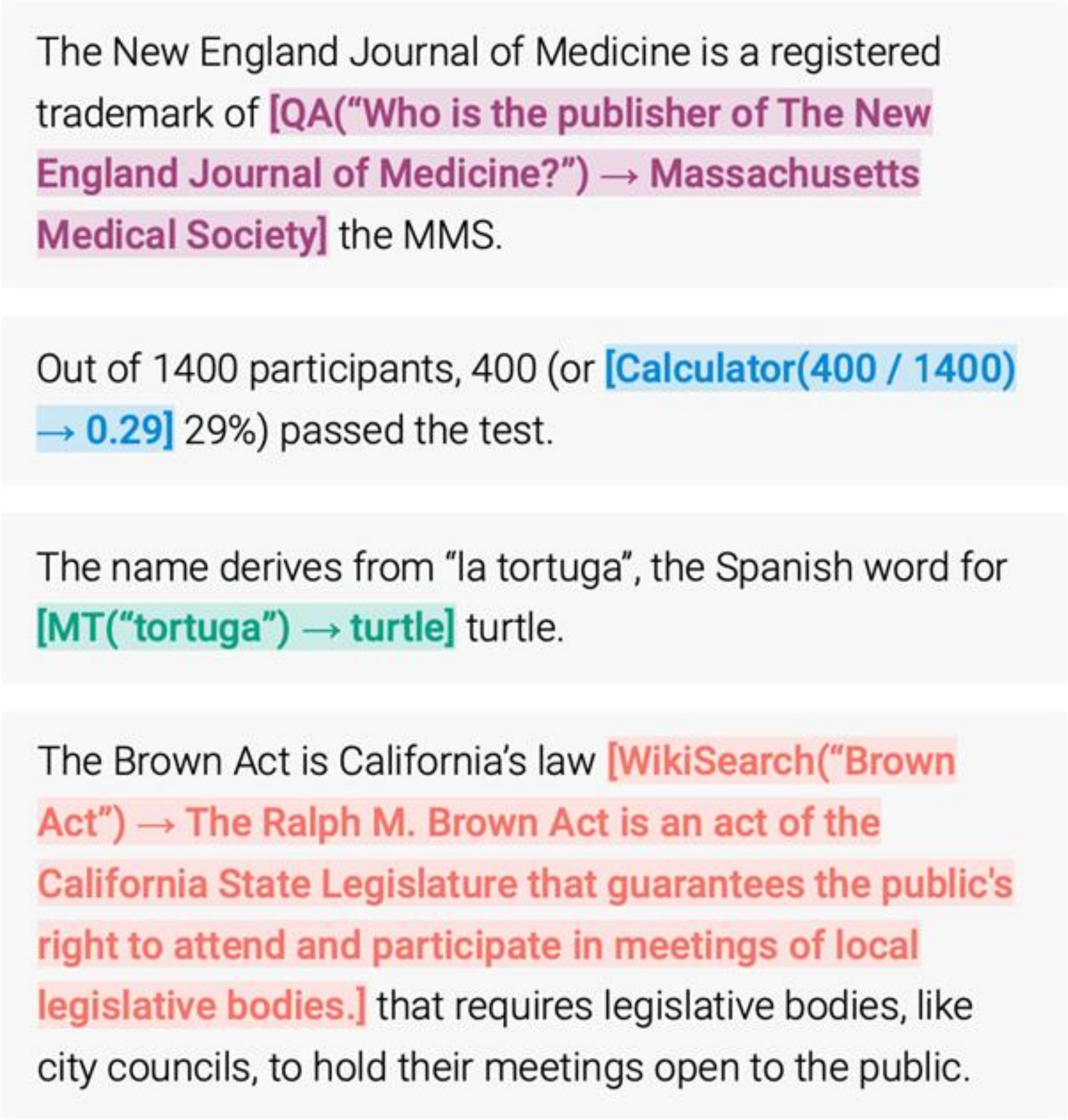
$\mathbf{x}^*$ = Pittsburgh is also known as [QA(What ...? → Steel City)] the Steel City.

# Toolformer Training

***Step 1:***

*Prompting to annotate potential API calls.*

Ask a pre-trained LM to annotate a dataset via few-shot learning with API call usage examples.



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.

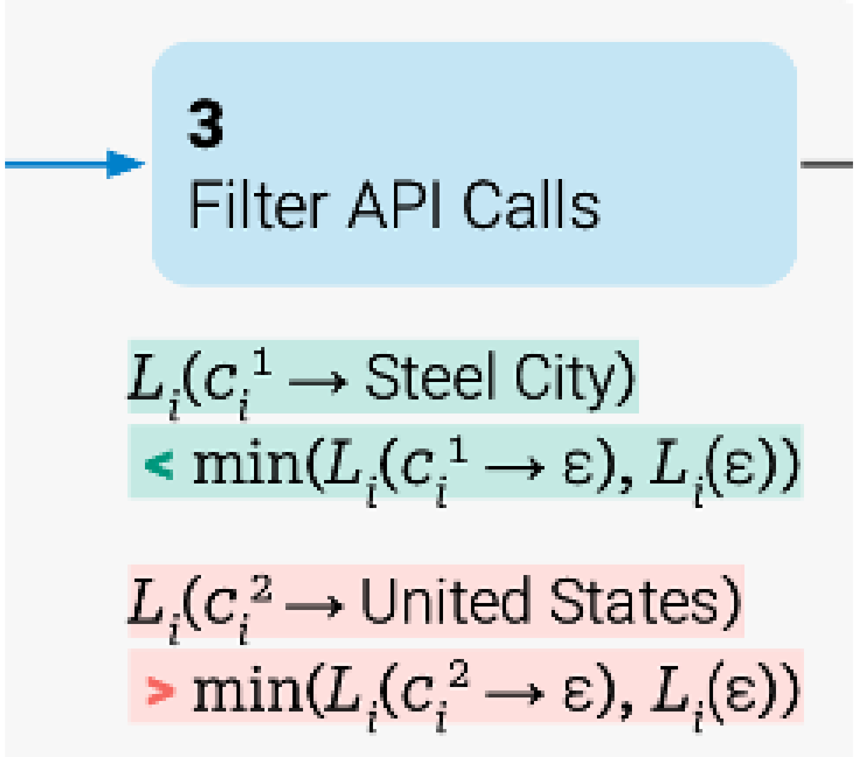
The Brown Act is California's law **[WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.]** that requires legislative bodies, like city councils, to hold their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

# Toolformer Training

***Step 2:***

*Filter annotations based on whether API calls help predict future tokens.*

Use a self-supervised loss to decide which API calls are actually helpful.



**3**

Filter API Calls

$L_i(c_i^1 \rightarrow \text{Steel City})$

$< \min(L_i(c_i^1 \rightarrow \varepsilon), L_i(\varepsilon))$

$L_i(c_i^2 \rightarrow \text{United States})$

$> \min(L_i(c_i^2 \rightarrow \varepsilon), L_i(\varepsilon))$

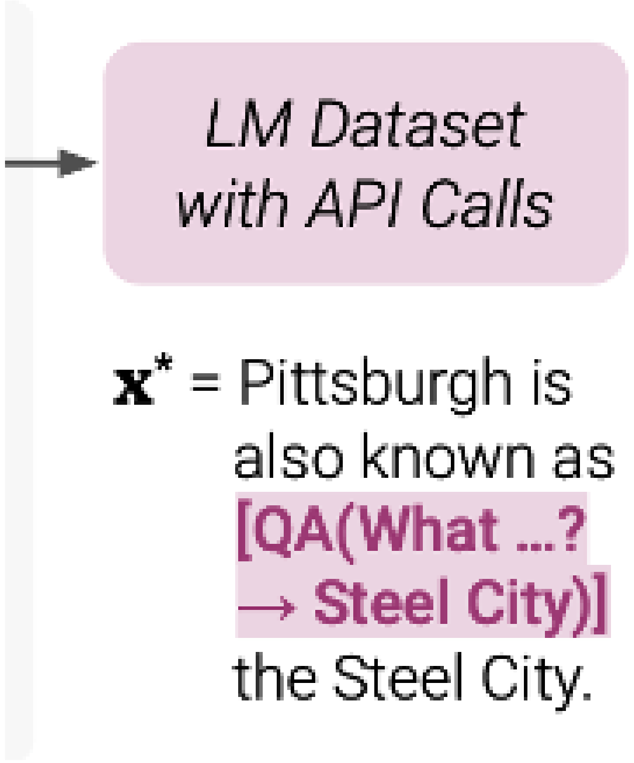
$$
\begin{aligned}
L_i^+ &= L_i(e(c_i, r_i)) \\
L_i^- &= \min(L_i(\varepsilon), L_i(e(c_i, \varepsilon)))
\end{aligned}
$$

# Toolformer Training

***Step 3:***

*Fine-tune LM on this annotated dataset.*

The training data is a combination of the original dataset and its augmented version.



$\mathbf{x}^*$ = Pittsburgh is also known as **[QA(What ...? → Steel City)]** the Steel City.

# Concluding Remarks

- On alignment
  - Training LLMs on various tasks enhances their ability to execute instructions for new tasks.
  - Instruction-tuned LLMs exhibits better zero-shot and few-shot capability.
  - The number of instruction tuning clusters and the scale of the model are important to the performance of LLMs.
  - Many new benchmarks have been developed to test the models capability as well as to alert their pitfalls.
- On augmentation
  - The emergence of reasoning abilities in LLMs facilitates the incorporation of various tools and knowledge bases, thereby greatly enhancing their overall capability.
  - Deciding which tools to use and what relevant external knowledge to integrate is the key.
  - As tasks get more complex, LLMs' long-term planning capability is put to test.