University of Ljubljana, Faculty of Computer and Information Science

Generative large language models

Prof Dr Marko Robnik-Šikonja

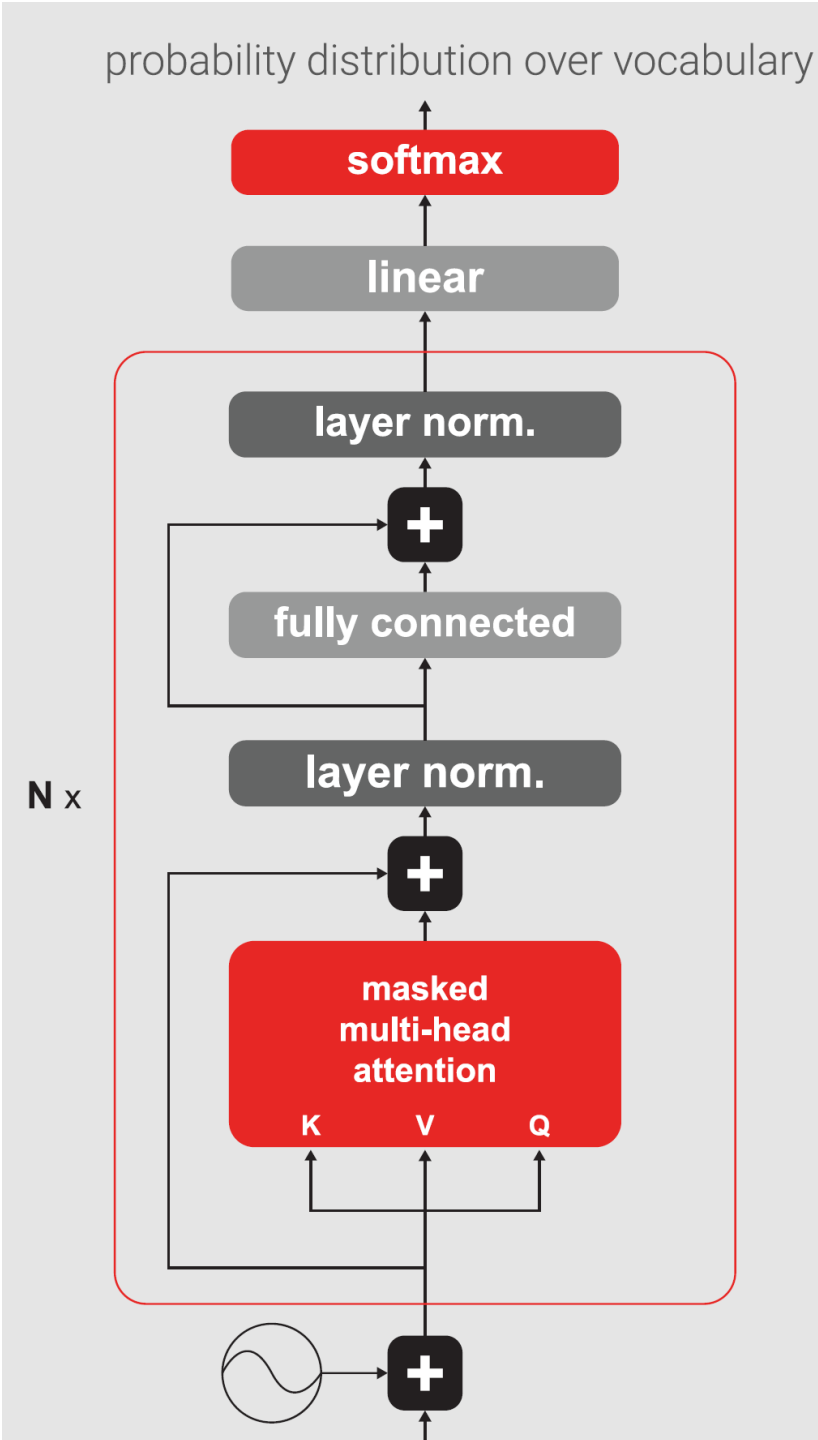
Natural Language Processing, Edition 2025

Contents

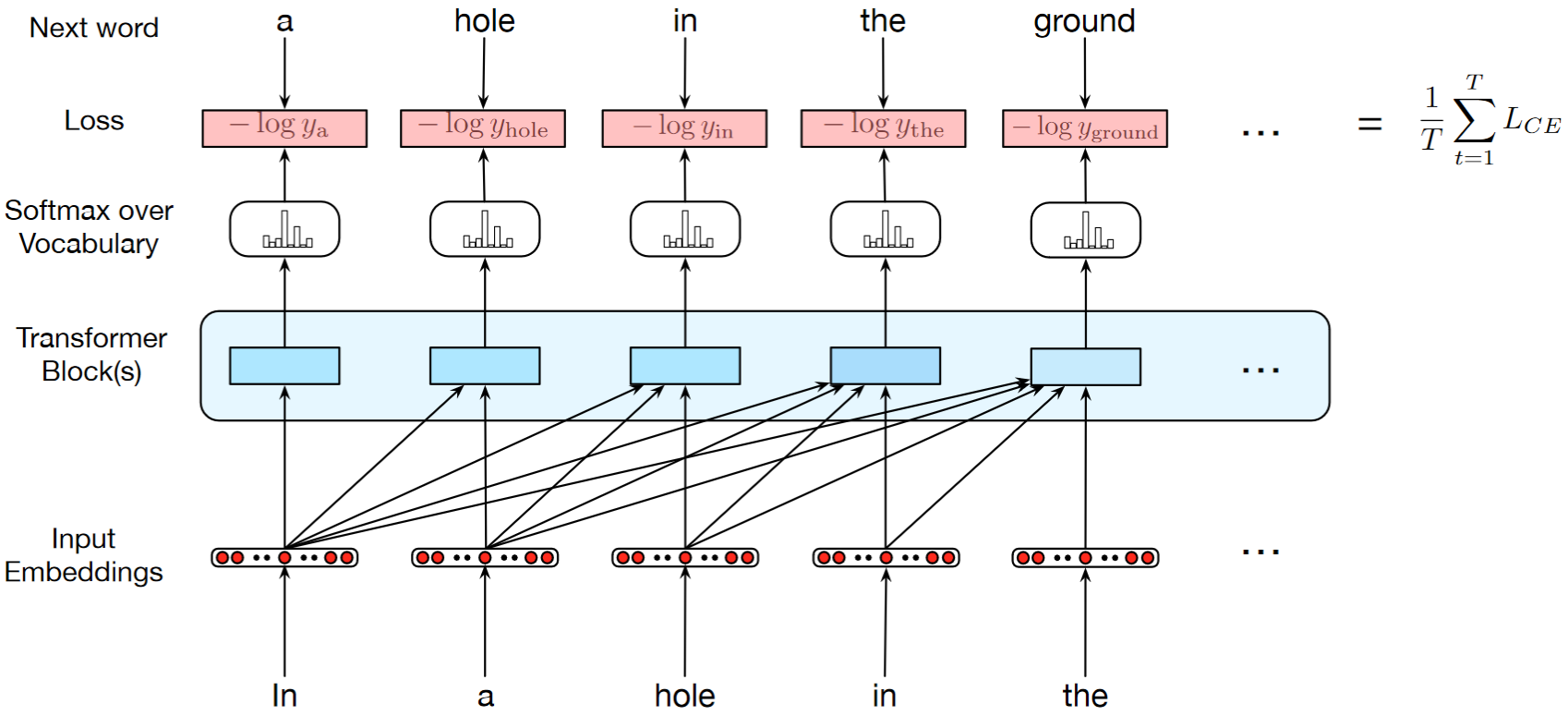
- GPT models
- training chat models
- prompt engineering
- retrieval augmented generation (RAG)

GPT family

- GPT: Generative Pre-trained Transformers
- use only the decoder part of transformer
- pretrained for language modeling (predicting the next word given the context)
- Shortcoming: unidirectional, does not incorporate bidirectionality
- “What are those?” he said while looking at my crocs.



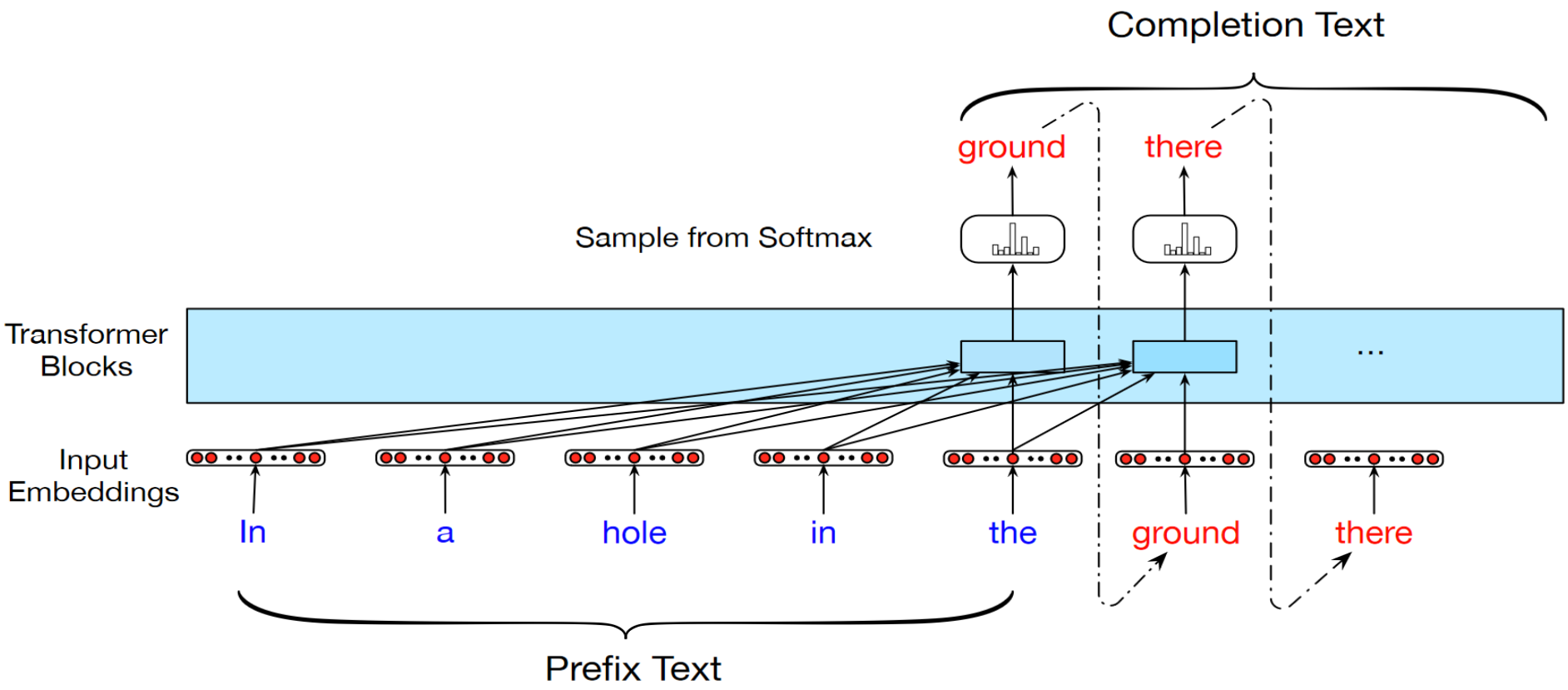
Transformer as a language model



- Can be computed in parallel

Autoregressive generators

- priming the generator with the context



- can be used also in summarization, QA and other generative tasks

GPT-2 and GPT-3

- few architectural changes, layer norm now applied to input of each subblock
- GPT-3 also uses some sparse attention layers
- more data, larger batch sizes (GPT-3 uses batch size of 3.2M)
- the models are scaled:

GPT-2:
48 layers, 25 heads
d_m = 1600, d = 64
context size = 1024
~ 1.5B parameters

GPT-3:
96 layers, 96 heads
d_m = 12288, d = 128
context size = 2048
~ 175B parameters

[1] Radford et al.: Language Models are Unsupervised Multitask Learners, 2019.
[2] Brown et al.: Language Models are Few-Shot Learners, 2020.

- see demos at https://transformer.huggingface.co/

In-context learning in GPT-2 and GPT-3

- GPT-2 and GPT-3 upgrade the “pre-train and fine-tune” training paradigm of GPT;
- GPT-2 explores unsupervised zero-shot learning, whereas in GPT-3 the authors expand the idea into in-context learning;
- use text input to condition the model on task description and some examples with ground truth.
- Uses zero-shot learning, one-shot learning, few-shot learning (as many examples as they can fit into the context, usually 10-100)
- no gradient updates are performed.

In-context learning

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:        ← task description
cheese =>                           ← prompt
```

One-shot

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

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← example
cheese =>                           ← prompt
```

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:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée        ←
plush girafe => girafe peluche      ←
cheese =>                           ← prompt
```

Figure source: Brown et al.: Language Models are Few-Shot Learners, 2020.

- GPT-3 is still a language model and can be used for text generation

- only 12% of respondents correctly classified this as not written by a human

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: **After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.**
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

Huge generative language models

- ChatGPT, OpenAI, Nov. 2022
  based on GPT-3.5 with additional training for dialogue
- uses RLHF (reinforcement learning with human feedback)
- demo: https://chat.openai.com/
- huge public impact, possibly disruptive for writing professions, learning, teaching, scientific writing
- GPT-4, 2023: even larger, allows longer context, image input

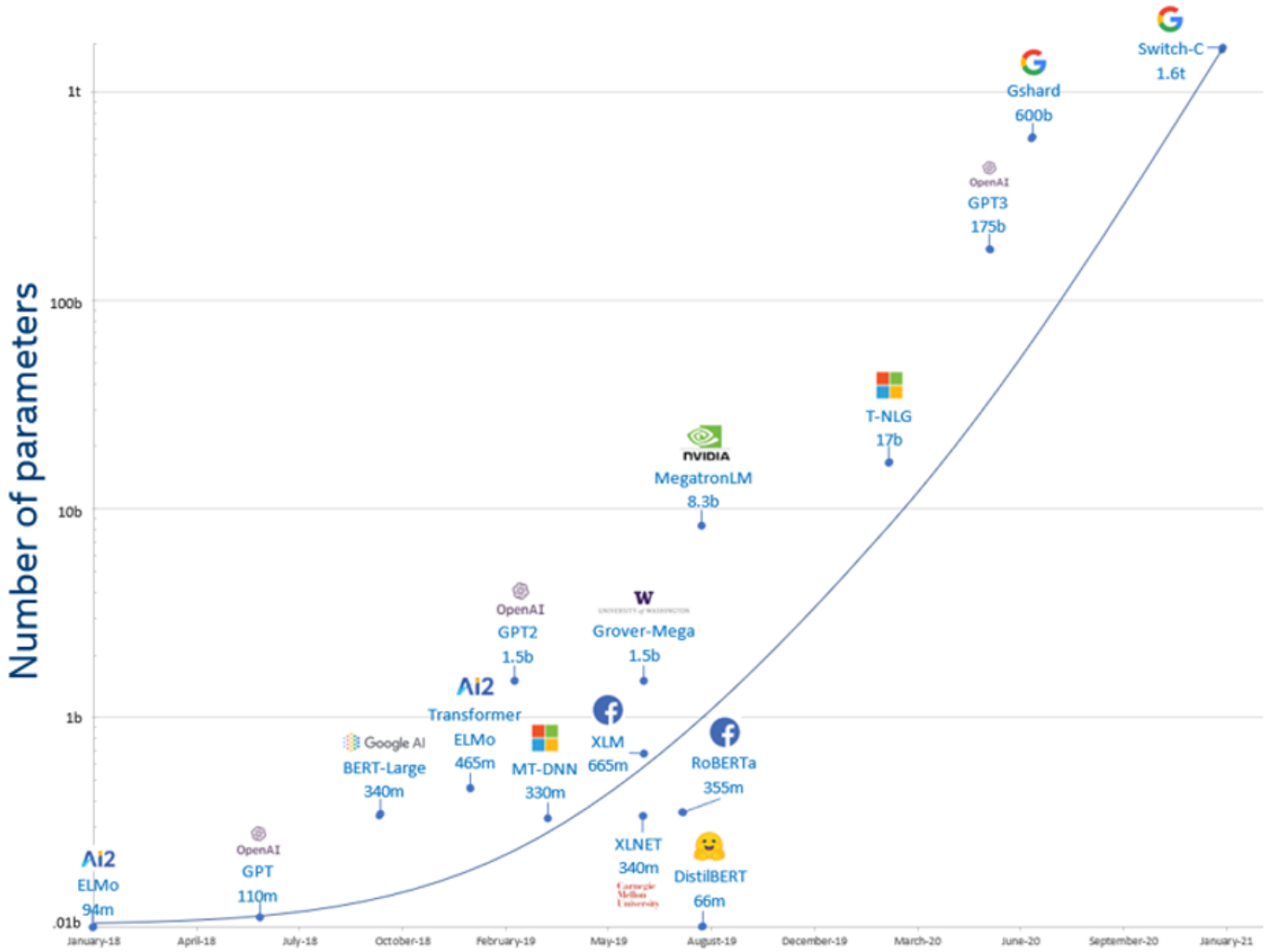


Figure 1: Exponential growth of number of parameters in DL models

LLMs are data hungry

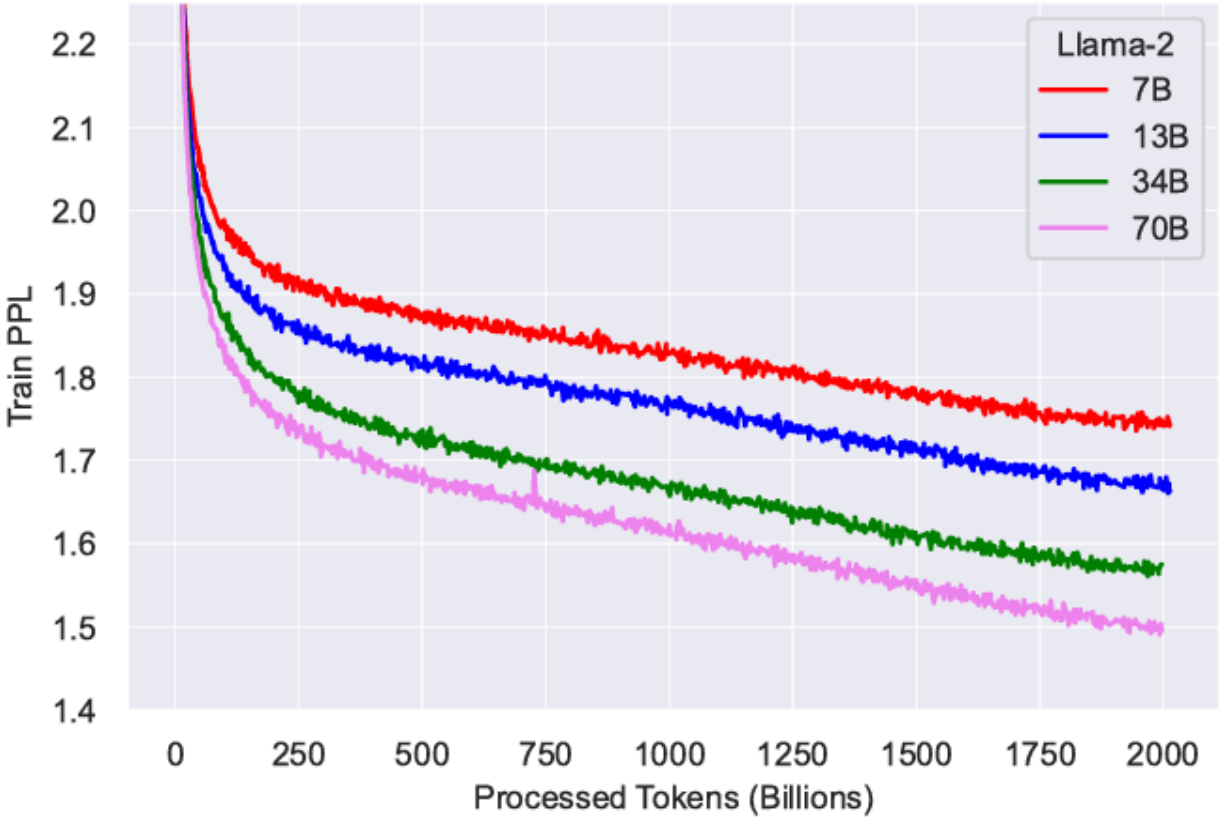


Figure 5: Training Loss for Llama 2 models. We compare the training loss of the Llama 2 family of models. We observe that after pretraining on 2T Tokens, the models still did not show any sign of saturation.

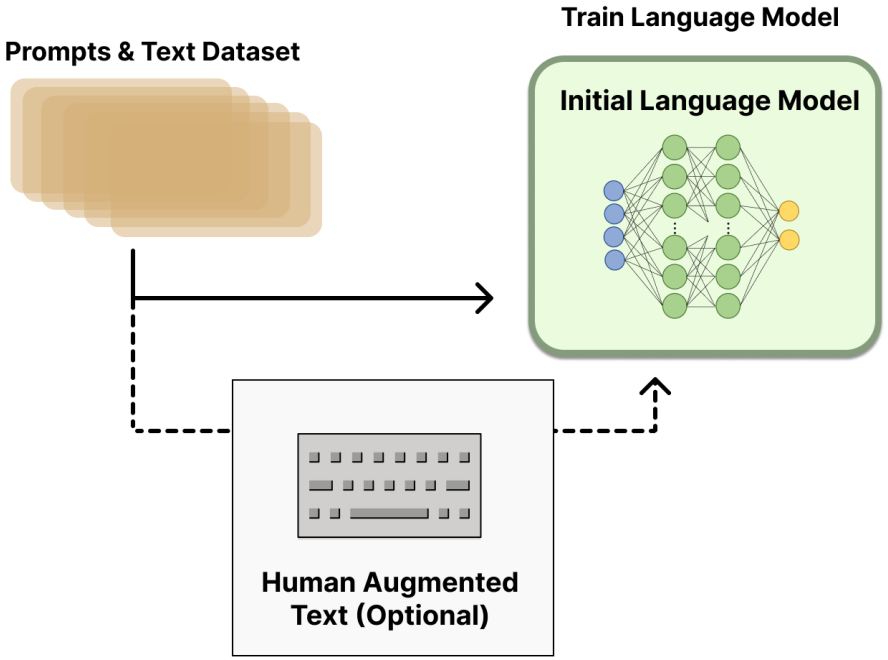
RLHF: idea

- Reinforcement Learning with Human Feedback
- A problem: Human feedback is not present during training
- Idea: Train a separate model on human feedback, this model can generate a reward to be used during training of LLM
- Three stages:
  1. Pretraining a language model (LM),
  2. Gathering data and training a reward model, and
  3. Fine-tuning the LM with reinforcement learning.

Nathan Lambert, Louis Castricato, Leandro von Werra, Alex Havrilla: "Illustrating Reinforcement Learning from Human Feedback (RLHF)", Hugging Face Blog, 2022.

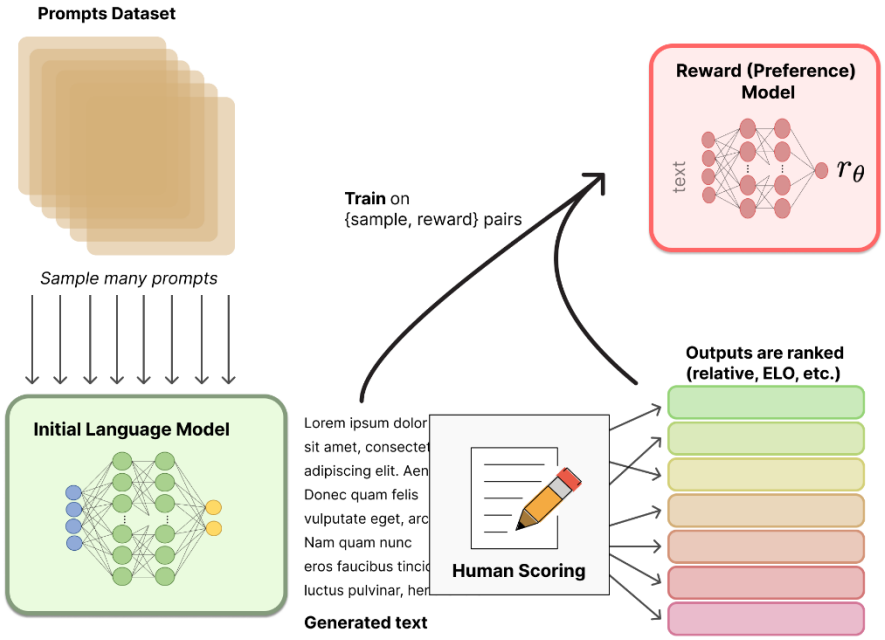
RLHF: the reward model 1/2

- input: a sequence of text, e.g., produced by LM and optionally improved by humans
- output: a scalar reward, representing the human preference of the text (e.g., a rank of the answer; or just a binary preference, which is even easier to obtain)
- the reward model could be an end-to-end LM, or the model ranks outputs, and the ranking is converted to reward



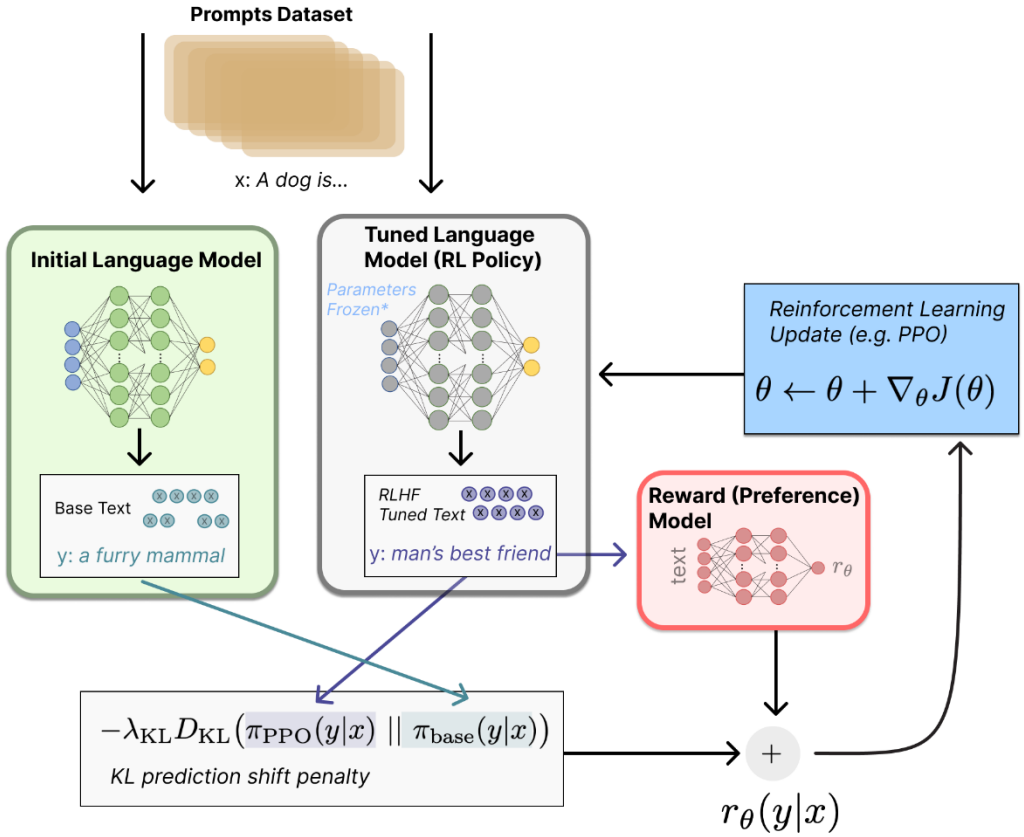
RLHF: the reward model 2/2

- the training dataset are pairs of prompts and (human improved) LM responses, e.g., 50k instances
- humans rank the responses instead of producing the direct reward as this produces better calibrated scores



RLHF: fine-tuning with RL

- RL does not change all parameters, most of parameters are frozen
- typical algorithm: Proximal Policy Optimization (PPO)



Prompt engineering (PE)

- Prompt engineering is a relatively new discipline for developing and optimizing prompts to efficiently use LLMs for a wide variety of tasks.
- Prompt engineering can help design robust and effective prompting techniques that interface with LLMs and other tools
- PE is a set of soft guidelines on how to design effective prompts
- PE is becoming an automatic approach to design effective prompts
- See https://www.promptingguide.ai/

PE basics

- Prompt engineering is based on the attention mechanism and relatively long input of LLMs
- Humans typically produce relatively short prompts
- A good prompt shall provide sufficient context
- A small variance in prompt can produce large variance in results
- PE is somewhere between engineering and guessing what might work
- However: certain well-established approaches exist, like Retrieval Augmented Generation (RAG)

Basic prompts

- What is wrong with the following prompt?
- Prompt

The sky is

- Output:

blue.

- Intention, context?

Basic prompts

- Better?

Complete the sentence:

The sky is

- Output:

blue during the day and dark at night.

- Still zero-shot

Typical prompt elements

- **Instruction** - a specific task or instruction you want the model to perform
- **Context** - external information or additional context that can steer the model to better responses
- **Input Data** - the input or question that we are interested to find a response for
- **Output Indicator** - the type or format of the output.

Example

Classify the text into neutral, negative, or positive

Text: I think the food was okay.

Sentiment:

- No context here, but what would be appropriate?

General guidelines for manual prompt construction

- designing prompts is an iterative process that requires a lot of experimentation to get good results
- start with simple prompts and keep adding more elements and context
- in many cases specificity, simplicity, and conciseness will give better results.
- For big task with many different subtasks, try to break down the task into simpler subtasks and keep building up as you get better results.

Specificity

- Prompt:

Extract the name of places in the following text.

Desired format:

Place: <comma_separated_list_of_places>

Input: "Although these developments are encouraging to researchers, much is still a mystery. “We often have a black box between the brain and the effect we see in the periphery,” says Henrique Veiga-Fernandes, a neuroimmunologist at the Champalimaud Centre for the Unknown in Lisbon. “If we want to use it in the therapeutic context, we actually need to understand the mechanism.“"

- Output:

Place: Champalimaud Centre for the Unknown, Lisbon

- Make prompts descriptive and detailed
- Provide examples in the prompt

Say what to do, avoid saying what not to do

- Note: LLMs are not good at negation
- Prompt:

The following is an agent that recommends movies to a customer. DO NOT ASK FOR INTERESTS. DO NOT ASK FOR PERSONAL INFORMATION.

Customer: Please recommend a movie based on my interests.
Agent:

- Output:

Sure, I can recommend a movie based on your interests. What kind of movie would you like to watch? Do you prefer action, comedy, romance, or something else?

- A better prompt:

The following is an agent that recommends movies to a customer. The agent is responsible to recommend a movie from the top global trending movies. It should refrain from asking users for their preferences and avoid asking for personal information. If the agent doesn't have a movie to recommend, it should respond "Sorry, couldn't find a movie to recommend today.".

Customer: Please recommend a movie based on my interests.

Agent:

- Output:

Sorry, I don't have any information about your interests. However, here's a list of the top global trending movies right now: [list of movies]. I hope you find something you like!

Few-shot prompting

- Few-shot prompting provides context to steer the model to better performance.

- Prompt:

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

- Output:

When we won the game, we all started to farduddle in celebration.

- Not only 1-shot, also 3-shot, 10-shot, etc.

Few-shot tips

- “The label space and the distribution of the input text specified by the demonstrations are both important (regardless of whether the labels are correct for individual inputs)"
- the format you use also plays a key role in performance, even if you just use random labels, this is much better than no labels at all.
- additional results show that selecting random labels from a true distribution of labels (instead of a uniform distribution) also helps.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H. and Zettlemoyer, L., 2022, December. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 11048-11064).

Example

- assigning random labels still works
- Prompt

This is awesome! // Negative
This is bad! // Positive
Wow that movie was rad! // Positive
What a horrible show! //

- Output:

Negative

Few-shot limitations

- providing examples is useful for solving some tasks
- if zero-shot prompting and few-shot prompting are not sufficient, we shall consider fine-tuning or using more advanced prompting techniques, like chain-of-thought reasoning

Zero-shot example

- Prompt

The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.

Answer:

- Output:

Yes, the odd numbers in this group add up to 107, which is an even number.

Few-shot example

- Prompt:

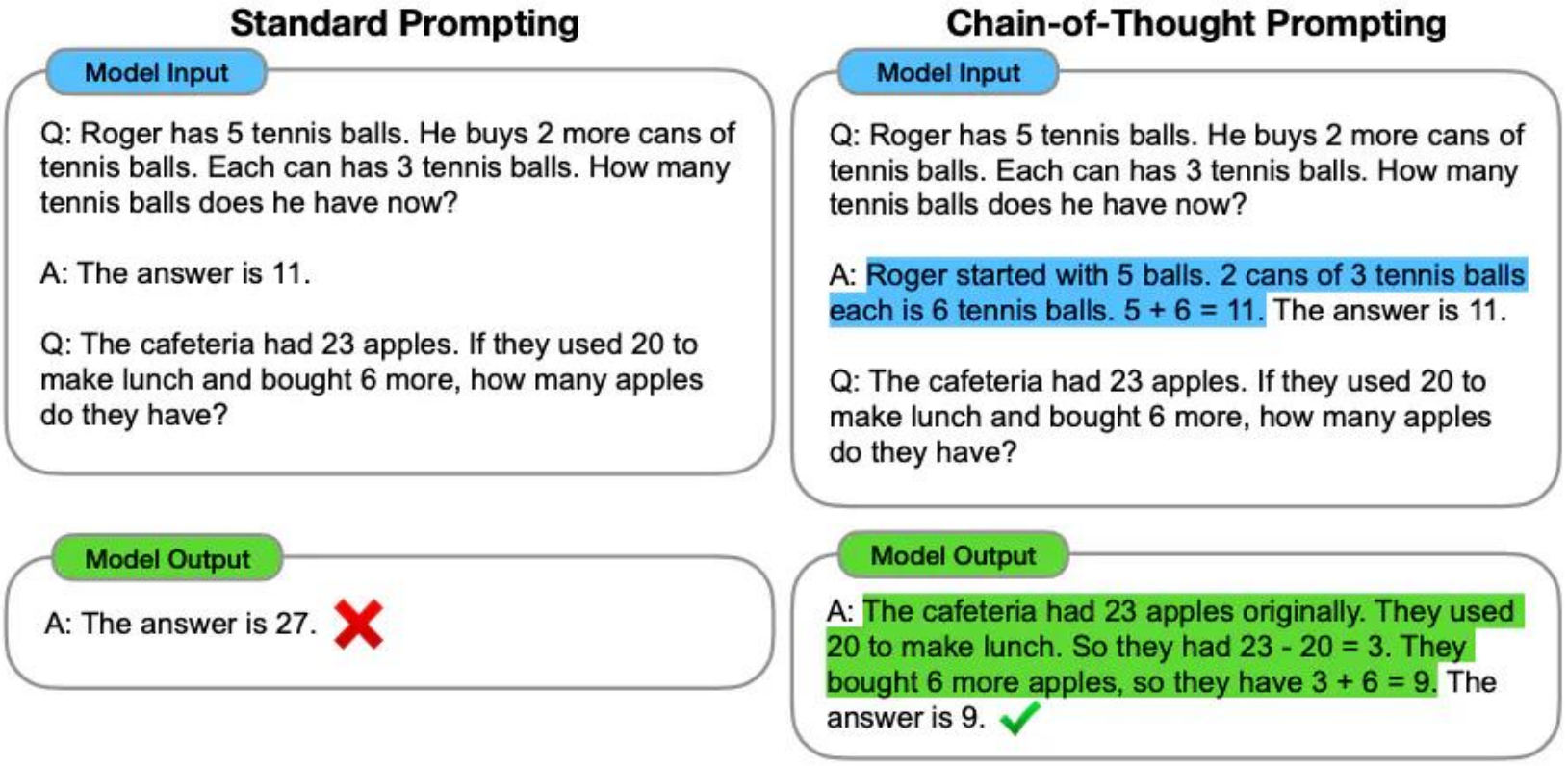
The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1.
A: The answer is False.
The odd numbers in this group add up to an even number: 17, 10, 19, 4, 8, 12, 24.
A: The answer is True.
The odd numbers in this group add up to an even number: 16, 11, 14, 4, 8, 13, 24.
A: The answer is True.
The odd numbers in this group add up to an even number: 17, 9, 10, 12, 13, 4, 2.
A: The answer is False.
The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.
A:

- Output:

The answer is True.

- The task requires several reasoning steps

Chain-of-Thought (CoT) prompting



Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V. and Zhou, D., 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, pp.24824-24837.

CoT example

- Prompt:

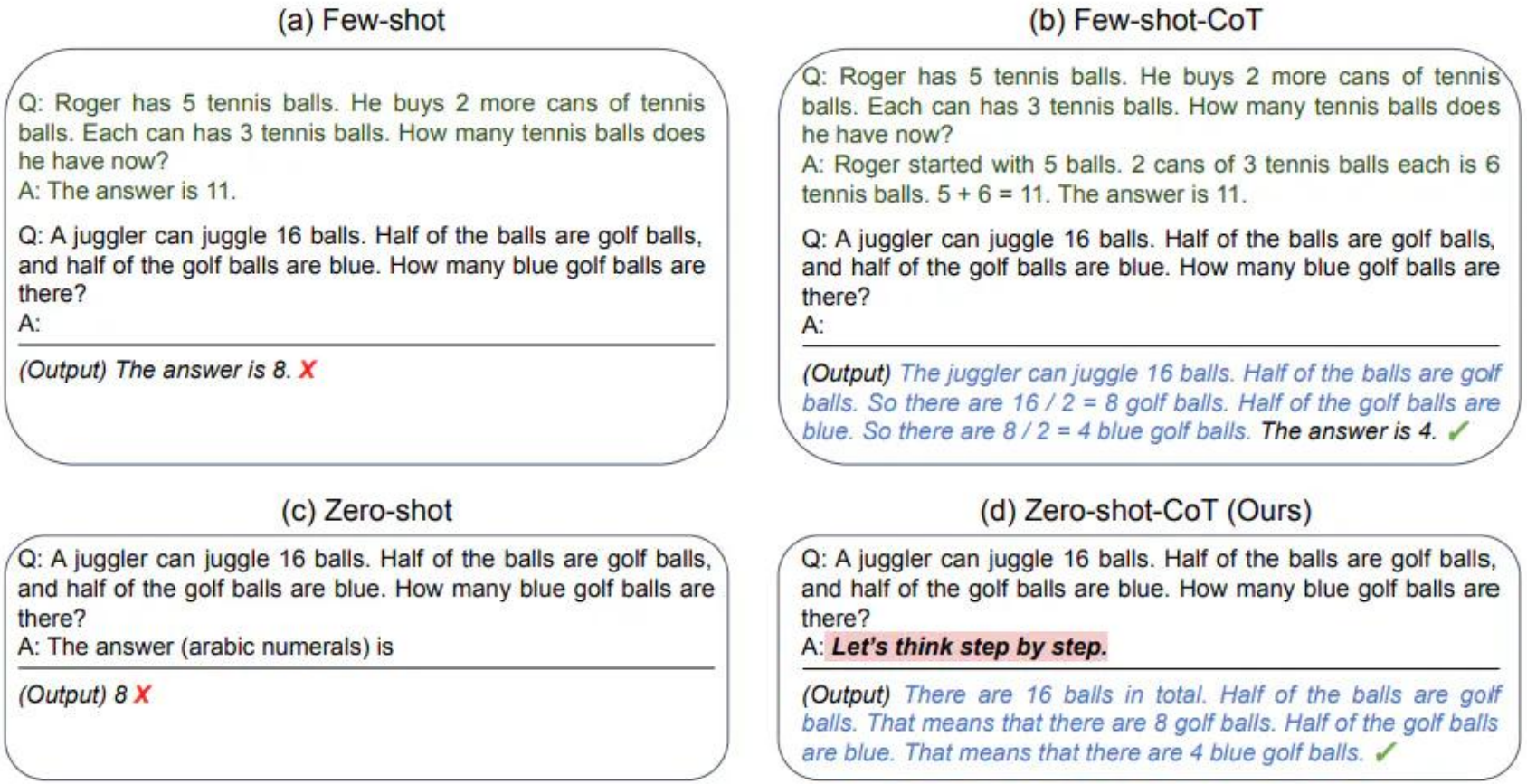
The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1.
A: Adding all the odd numbers (9, 15, 1) gives 25. The answer is False.
The odd numbers in this group add up to an even number: 17, 10, 19, 4, 8, 12, 24.
A: Adding all the odd numbers (17, 19) gives 36. The answer is True.
The odd numbers in this group add up to an even number: 16, 11, 14, 4, 8, 13, 24.
A: Adding all the odd numbers (11, 13) gives 24. The answer is True.
The odd numbers in this group add up to an even number: 17, 9, 10, 12, 13, 4, 2.
A: Adding all the odd numbers (17, 9, 13) gives 39. The answer is False.
The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.
A:

- Output:

Adding all the odd numbers (15, 5, 13, 7, 1) gives 41. The answer is False.

- Works even with a single expanded example

Zero-shot COT Prompting



- this special prompt seems to work also in many other cases

Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. and Iwasawa, Y., 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, *35*, pp.22199-22213.

Automatic Chain-of-Thought (Auto-CoT)

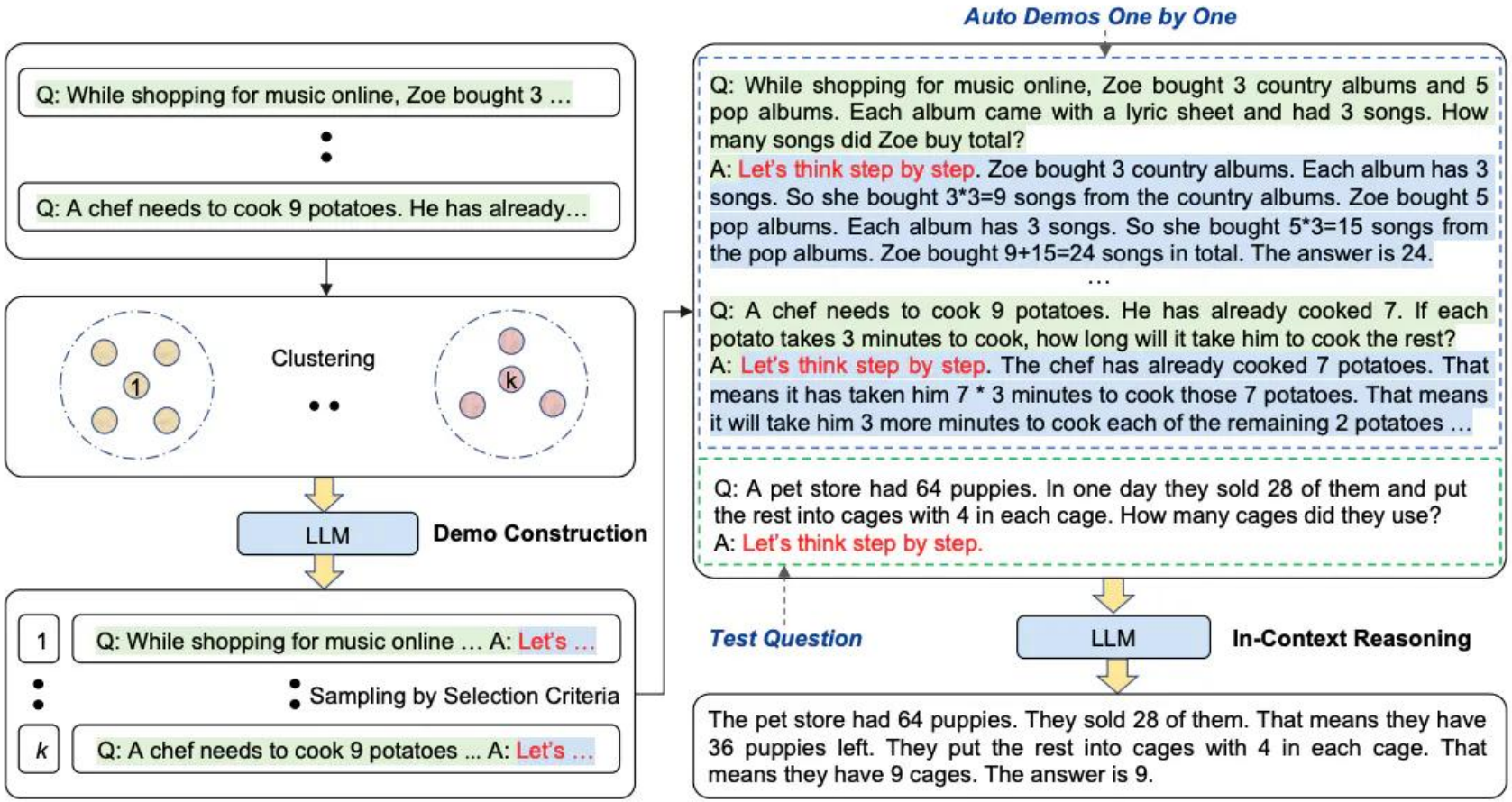
- CoT with demonstrations requires hand-crafting effective and diverse examples.
- might not always work
- let us use eliminate manual work by using LLMs with "Let's think step by step" prompt to generate reasoning chains for demonstrations one by one.
- This process can still produce mistakes in generated chains.
- To reduce the effects of the mistakes, produce diverse demonstrations
- Auto-CoT consists of two main stages:

  1) **question clustering**: partition questions of a given dataset into a few clusters

  2) **demonstration sampling**: select a representative question from each cluster and generate its reasoning chain using Zero-Shot-CoT with simple heuristics, like length of questions (e.g., 60 tokens) and number of steps in rationale (e.g., 5 reasoning steps). This encourages the model to use simple and accurate demonstrations.

Zhang, Z., Zhang, A., Li, M. and Smola, A., 2022. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations*.

Auto-CoT


Q: While shopping for music online, Zoe bought 3 …
Q: A chef needs to cook 9 potatoes. He has already…
Clustering
1
k
LLM
Demo Construction
1
Q: While shopping for music online … A: Let's …
Sampling by Selection Criteria
k
Q: A chef needs to cook 9 potatoes ... A: Let's …
Auto Demos One by One
Q: While shopping for music online, Zoe bought 3 country albums and 5 pop albums. Each album came with a lyric sheet and had 3 songs. How many songs did Zoe buy total?
A: Let's think step by step. Zoe bought 3 country albums. Each album has 3 songs. So she bought 3*3=9 songs from the country albums. Zoe bought 5 pop albums. Each album has 3 songs. So she bought 5*3=15 songs from the pop albums. Zoe bought 9+15=24 songs in total. The answer is 24.
…
Q: A chef needs to cook 9 potatoes. He has already cooked 7. If each potato takes 3 minutes to cook, how long will it take him to cook the rest?
A: Let's think step by step. The chef has already cooked 7 potatoes. That means it has taken him 7 * 3 minutes to cook those 7 potatoes. That means it will take him 3 more minutes to cook each of the remaining 2 potatoes …
Q: A pet store had 64 puppies. In one day they sold 28 of them and put the rest into cages with 4 in each cage. How many cages did they use?
A: Let's think step by step.
Test Question
LLM
In-Context Reasoning
The pet store had 64 puppies. They sold 28 of them. That means they have 36 puppies left. They put the rest into cages with 4 in each cage. That means they have 9 cages. The answer is 9.

Self-Consistency with CoT

- replaces the greedy decoding used in chain-of-thought prompting
- produce multiple, diverse reasoning paths through few-shot CoT, and use several generations to select the most consistent answer.
- improves CoT prompting on arithmetic and commonsense reasoning tasks.

Wang, X., Wei, J., Schuurmans, D., Le, Q.V., Chi, E.H., Narang, S., Chowdhery, A. and Zhou, D., 2022, September. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In The Eleventh International Conference on Learning Representations.

Zero shot example

- Prompt:

When I was 6 my sister was half my age. Now I'm 70 how old is my sister?

- Output:

35

- Let us try self-consistency

Self-consistency example

- Prompt:

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A: We start with 15 trees. Later we have 21 trees. The difference must be the number of trees they planted. So, they must have planted 21 - 15 = 6 trees. The answer is 6.

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

A: There are 3 cars in the parking lot already. 2 more arrive. Now there are 3 + 2 = 5 cars. The answer is 5.

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

A: Leah had 32 chocolates and Leah's sister had 42. That means there were originally 32 + 42 = 74 chocolates. 35 have been eaten. So in total they still have 74 - 35 = 39 chocolates. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?

A: Jason had 20 lollipops. Since he only has 12 now, he must have given the rest to Denny. The number of lollipops he has given to Denny must have been 20 - 12 = 8 lollipops. The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?

A: He has 5 toys. He got 2 from mom, so after that he has 5 + 2 = 7 toys. Then he got 2 more from dad, so in total he has 7 + 2 = 9 toys. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: There are 4 days from monday to thursday. 5 computers were added each day. That means in total 4 * 5 = 20 computers were added. There were 9 computers in the beginning, so now there are 9 + 20 = 29 computers.

The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?

A: Michael initially had 58 balls. He lost 23 on Tuesday, so after that he has 58 - 23 = 35 balls. On Wednesday he lost 2 more so now he has 35 - 2 = 33 balls. The answer is 33.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?

A: She bought 5 bagels for $3 each. This means she spent $15. She has $8 left.

Q: When I was 6 my sister was half my age. Now I'm 70 how old is my sister?

A:

- Output 1:

When I was 6 my sister was half my age, so she was 3. Now I am 70, so she is 70 - 3 = 67. The answer is 67.

- Output 2:

When the narrator was 6, his sister was half his age, which is 3. Now that the narrator is 70, his sister would be 70 - 3 = 67 years old. The answer is 67.

- Output 3:

When I was 6 my sister was half my age, so she was 3. Now I am 70, so she is 70/2 = 35. The answer is 35.

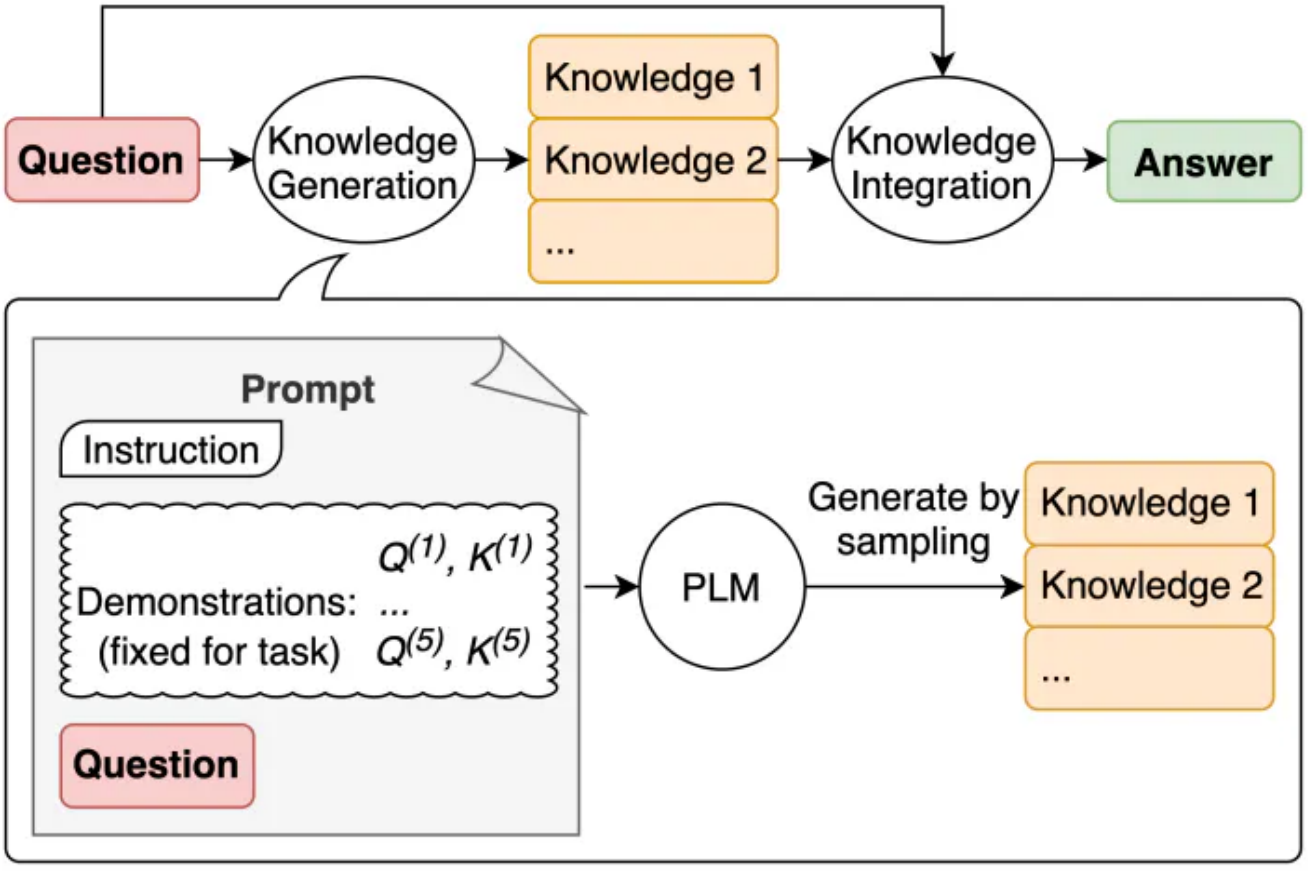
- Computing for the final answer involves a few steps. In the example, a majority answer is already emerging.

Generated knowledge prompting

- RAG tries to incorporate knowledge or information into the prompt to help the model make better predictions.
- A similar idea uses LLM to generate knowledge before making a prediction.
- The generated knowledge is used as part of the prompt

Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Le Bras, R., Choi, Y. and Hajishirzi, H., 2022, May. Generated Knowledge Prompting for Commonsense Reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 3154-3169).

Generated knowledge prompting



Zero-shot example

- Prompt:

Part of golf is trying to get a higher point total than others. Yes or No?

- Output:

Yes.

- Can we improve this with knowledge generation?
- First, we generate a few "knowledge“ pieces.

Example: knowledge generation prompt

- Prompt:

Input: Greece is larger than Mexico.

Knowledge: Greece is approximately 131,957 sq km, while Mexico is approximately 1,964,375 sq km, making Mexico 1,389% larger than Greece.

Input: Glasses always fog up.

Knowledge: Condensation occurs on eyeglass lenses when water vapor from your sweat, breath, and ambient humidity lands on a cold surface, cools, and then changes into tiny drops of liquid, forming a film that you see as fog. Your lenses will be relatively cool compared to your breath, especially when the outside air is cold.

Input: A fish is capable of thinking.

Knowledge: Fish are more intelligent than they appear. In many areas, such as memory, their cognitive powers match or exceed those of 'higher' vertebrates including non-human primates. Fish's long-term memories help them keep track of complex social relationships.

Input: A common effect of smoking lots of cigarettes in one's lifetime is a higher than normal chance of getting lung cancer.

Knowledge: Those who consistently averaged less than one cigarette per day over their lifetime had nine times the risk of dying from lung cancer than never smokers. Among people who smoked between one and 10 cigarettes per day, the risk of dying from lung cancer was nearly 12 times higher than that of never smokers.

Input: A rock is the same size as a pebble.

Knowledge: A pebble is a clast of rock with a particle size of 4 to 64 millimetres based on the Udden-Wentworth scale of sedimentology. Pebbles are generally considered larger than granules (2 to 4 millimetres diameter) and smaller than cobbles (64 to 256 millimetres diameter).

Input: Part of golf is trying to get a higher point total than others.

Knowledge:

Example: knowledge generation

- Knowledge 1:

The objective of golf is to play a set of holes in the least number of strokes. A round of golf typically consists of 18 holes. Each hole is played once in the round on a standard golf course. Each stroke is counted as one point, and the total number of strokes is used to determine the winner of the game.

- Knowledge 2:

Golf is a precision club-and-ball sport in which competing players (or golfers) use many types of clubs to hit balls into a series of holes on a course using the fewest number of strokes. The goal is to complete the course with the lowest score, which is calculated by adding up the total number of strokes taken on each hole. The player with the lowest score wins the game.

Example: integrate the knowledge and get a prediction

- Prompt:

Question: Part of golf is trying to get a higher point total than others. Yes or No?

Knowledge: The objective of golf is to play a set of holes in the least number of strokes. A round of golf typically consists of 18 holes. Each hole is played once in the round on a standard golf course. Each stroke is counted as one point, and the total number of strokes is used to determine the winner of the game.

Explain and Answer:

- Answer 1 (confidence very high):

No, the objective of golf is not to get a higher point total than others. Rather, the objective is to play a set of holes in the least number of strokes. The total number of strokes is used to determine the winner of the game, not the total number of points.

Question: Part of golf is trying to get a higher point total than others. Yes or No?

Knowledge: Golf is a precision club-and-ball sport in which competing players (or golfers) use many types of clubs to hit balls into a series of holes on a course using the fewest number of strokes. The goal is to complete the course with the lowest score, which is calculated by adding up the total number of strokes taken on each hole. The player with the lowest score wins the game.

Explain and Answer:

- Answer 2 (confidence is a lot lower):

Yes, part of golf is trying to get a higher point total than others. Each player tries to complete the course with the lowest score, which is calculated by adding up the total number of strokes taken on each hole. The player with the lowest score wins the game

- A few more details to consider before arriving at the final answer

Prompt chaining

- To improve the reliability and performance of LLMs in complex tasks, break tasks into subtasks.
- identify subtasks, prompt LLM with a subtask, and use its response as input to another prompt
- Prompt chaining creates a chain of prompt operations
- Prompt chaining may boost the transparency of the LLM application, increases controllability, and reliability.
- Easier to debug problems with model responses
- Easier to analyze and improve performance in the different stages that need improvement.

Example of prompt chaining for document QA

- In answering questions about a large text document, it helps to design two different prompts where the first prompt is responsible for extracting relevant quotes to answer a question and a second prompt takes as input the quotes and original document to answer a given question.

- Prompt 1:

You are a helpful assistant. Your task is to help answer a question given in a document. The first step is to extract quotes relevant to the question from the document, delimited by ####. Please output the list of quotes using <quotes></quotes>. Respond with "No relevant quotes found!" if no relevant quotes were found.

####

{{document}}

####

Question: What are the prompting techniques mentioned in the document?

- in the example, {{document}} is an article from Wikipedia about prompt engineering: https://en.wikipedia.org/wiki/Prompt_engineering

Example: first response

- Output of Prompt 1:

<quotes>
- Chain-of-thought (CoT) prompting[27]
- Generated knowledge prompting[37]
- Least-to-most prompting[38]
- Self-consistency decoding[39]
- Complexity-based prompting[41]
- Self-refine[42]
- Tree-of-thought prompting[43]
- Maieutic prompting[45]
- Directional-stimulus prompting[46]
- Textual inversion and embeddings[59]
- Using gradient descent to search for prompts[61][62][63][64]
- Prompt injection[65][66][67]
</quotes>

- This response is used as an input to the second prompt

Example: second prompt

Given a set of relevant quotes (delimited by <quotes></quotes>) extracted from a document and the original document (delimited by ####), please compose an answer to the question. Ensure that the answer is accurate, has a friendly tone, and sounds helpful.

####
{{document}}
####

<quotes>
- Chain-of-thought (CoT) prompting[27]
- Generated knowledge prompting[37]
- Least-to-most prompting[38]
- Self-consistency decoding[39]
- Complexity-based prompting[41]
- Self-refine[42]
- Tree-of-thought prompting[43]
- Maieutic prompting[45]
- Directional-stimulus prompting[46]
- Textual inversion and embeddings[59]
- Using gradient descent to search for prompts[61][62][63][64]
- Prompt injection[65][66][67]
</quotes>

Example: output of prompt 2

The prompting techniques mentioned in the document include:

1. Chain-of-thought (CoT) prompting[27]
2. Generated knowledge prompting[37]
3. Least-to-most prompting[38]
4. Self-consistency decoding[39]
5. Complexity-based prompting[41]
6. Self-refine[42]
7. Tree-of-thought prompting[43]
8. Maieutic prompting[45]
9. Directional-stimulus prompting[46]
10. Textual inversion and embeddings[59]
11. Using gradient descent to search for prompts[61][62][63][64]
12. Prompt injection[65][66][67]

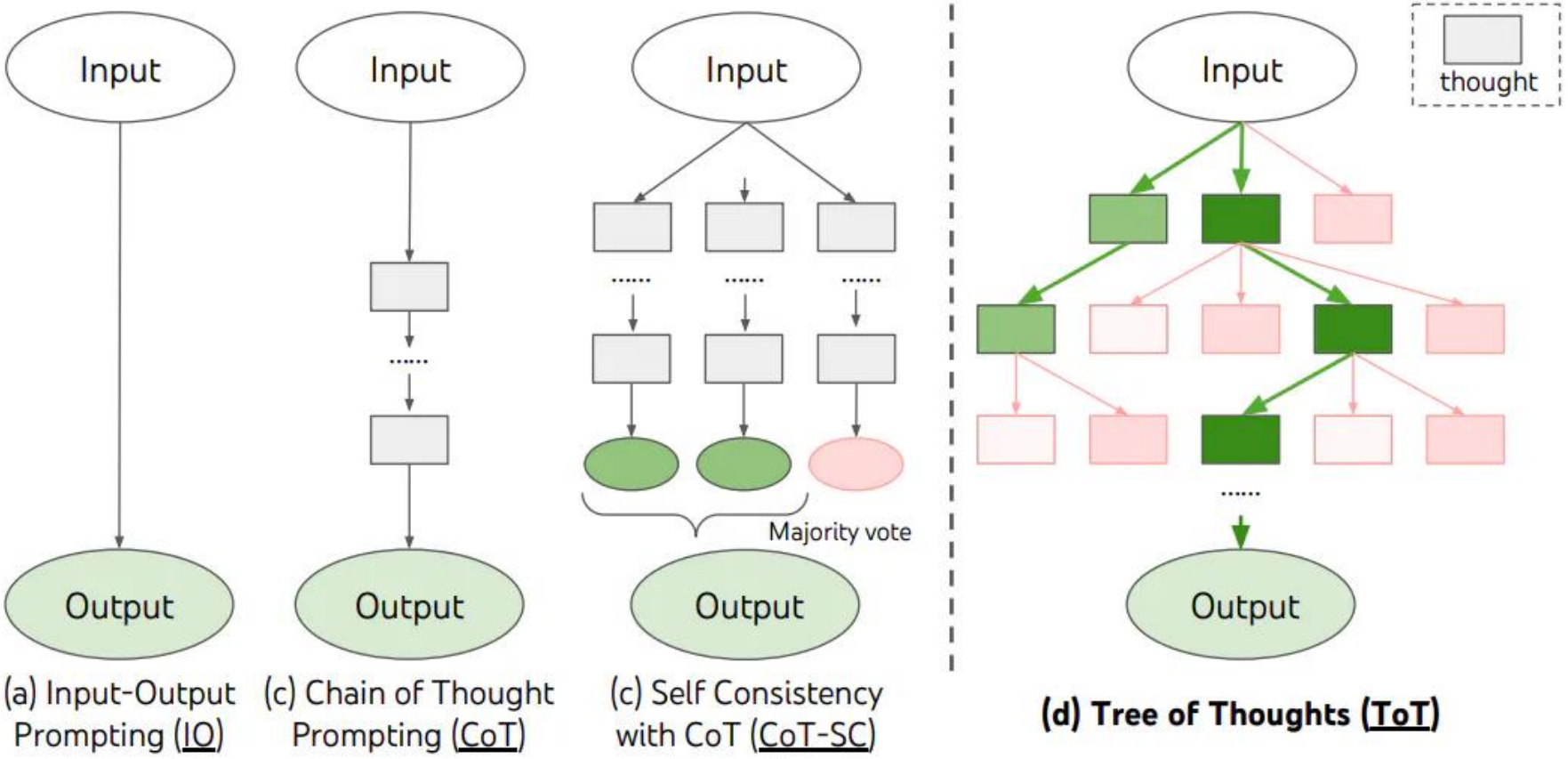
Each of these techniques employs unique strategies to enhance or specify the interactions with large language models to produce the desired outcomes

- additional work with cleaning the response, e.g., removing references

Tree of Thoughts (ToT)

- ToT explores over thoughts that serve as intermediate steps for general problem solving with language models.
- ToT maintains a tree of thoughts, where thoughts represent coherent language sequences that serve as intermediate steps toward solving a problem
- LLM self-evaluates the progress through intermediate thoughts made towards solving a problem through a reasoning process.
- LM's ability to generate and evaluate thoughts is then combined with search algorithms (e.g., breadth-first search and depth-first search) to enable systematic exploration of thoughts with lookahead and backtracking.

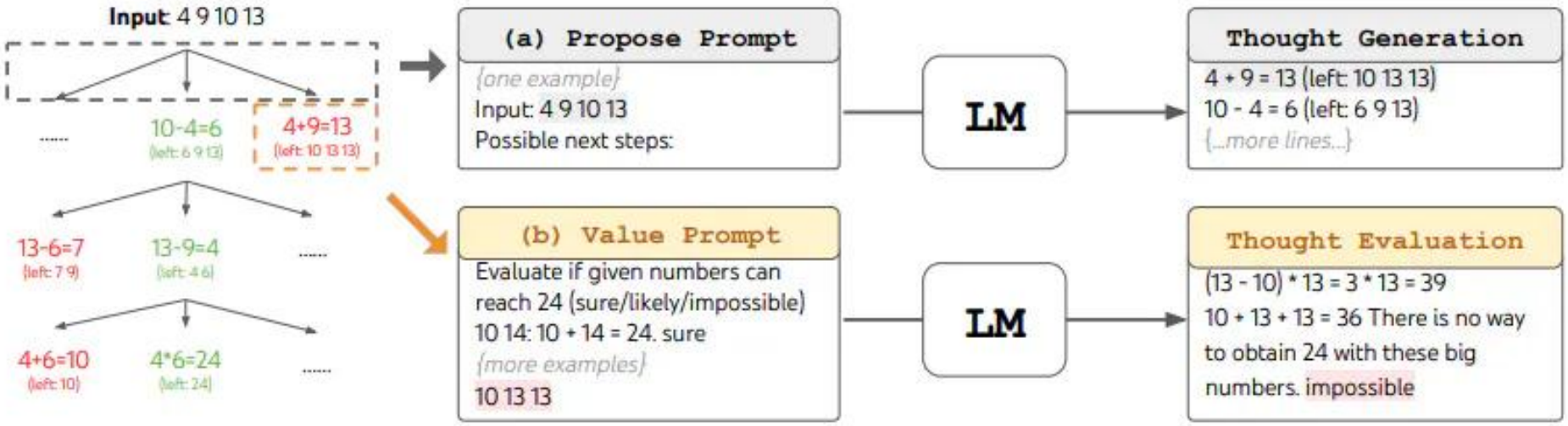
ToT comparison



ToT example

- Game of 24: Make the number 24 from the four numbers shown. You can add, subtract, multiply and divide. Use all four numbers on the card, but use each number only once. You do not have to use all four operations.
- different tasks require defining the number of candidates and the number of thoughts/steps.
- in Game of 24. the example requires decomposing the thoughts into 3 steps, each involving an intermediate equation. At each step, the best b=5 candidates are kept.

- To perform BFS, the LLM is prompted to evaluate each thought candidate as "sure/maybe/impossible" with regard to reaching 24, the aim is to promote correct partial solutions that can be predicted within few lookahead trials, and eliminate impossible partial solutions based on "too big/small" commonsense, and keep the rest "maybe"". Values are sampled 3 times for each thought.
- For this task, ToT is more successful than alternative prompting approaches



Alternative ToT formulation

- prompt the LLM to evaluate intermediate thoughts in a single prompt
- Prompt:

Imagine three different experts are answering this question.
All experts will write down 1 step of their thinking, then share it with the group.
Then all experts will go on to the next step, etc.
If any expert realises they're wrong at any point then they leave.
The question is:
Bob is in the living room.
He walks to the kitchen, carrying a cup.
He puts a ball in the cup and carries the cup to the bedroom.
He turns the cup upside down, then walks to the garden.
He puts the cup down in the garden, then walks to the garage.
Where is the ball?

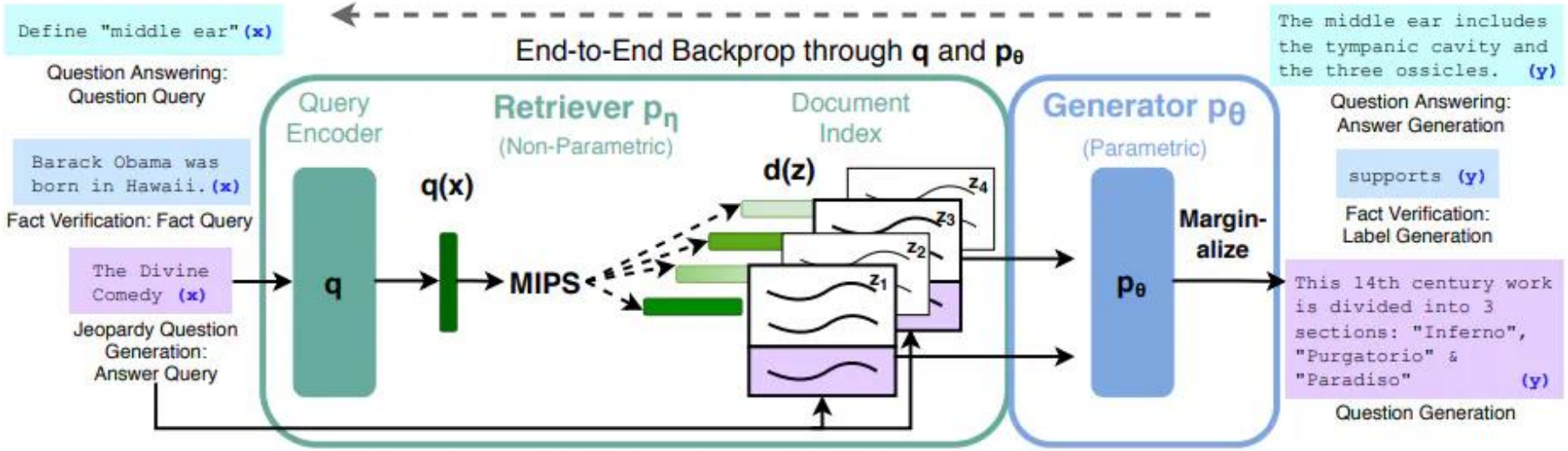
- Better results than using the question without breakdown instructions:

Dave Hulbert (2023). Using Tree-of-Thought Prompting to boost ChatGPT's reasoning. https://github.com/dave1010/tree-of-thought-prompting

Retrieval Augmented Generation (RAG)

- For complex and knowledge-intensive tasks, LLM accesses external knowledge sources to complete tasks.
- Improves factual consistency, reliability of the generated responses, reduces hallucinations
- RAG takes input and retrieves a set of relevant/supporting documents given a source (e.g., Wikipedia). The documents are concatenated as context with the original input prompt and used as the input to LLM which produces the final output.
- RAG adapts to dynamic situations (facts could evolve over time)
- successful in QA

RAG details



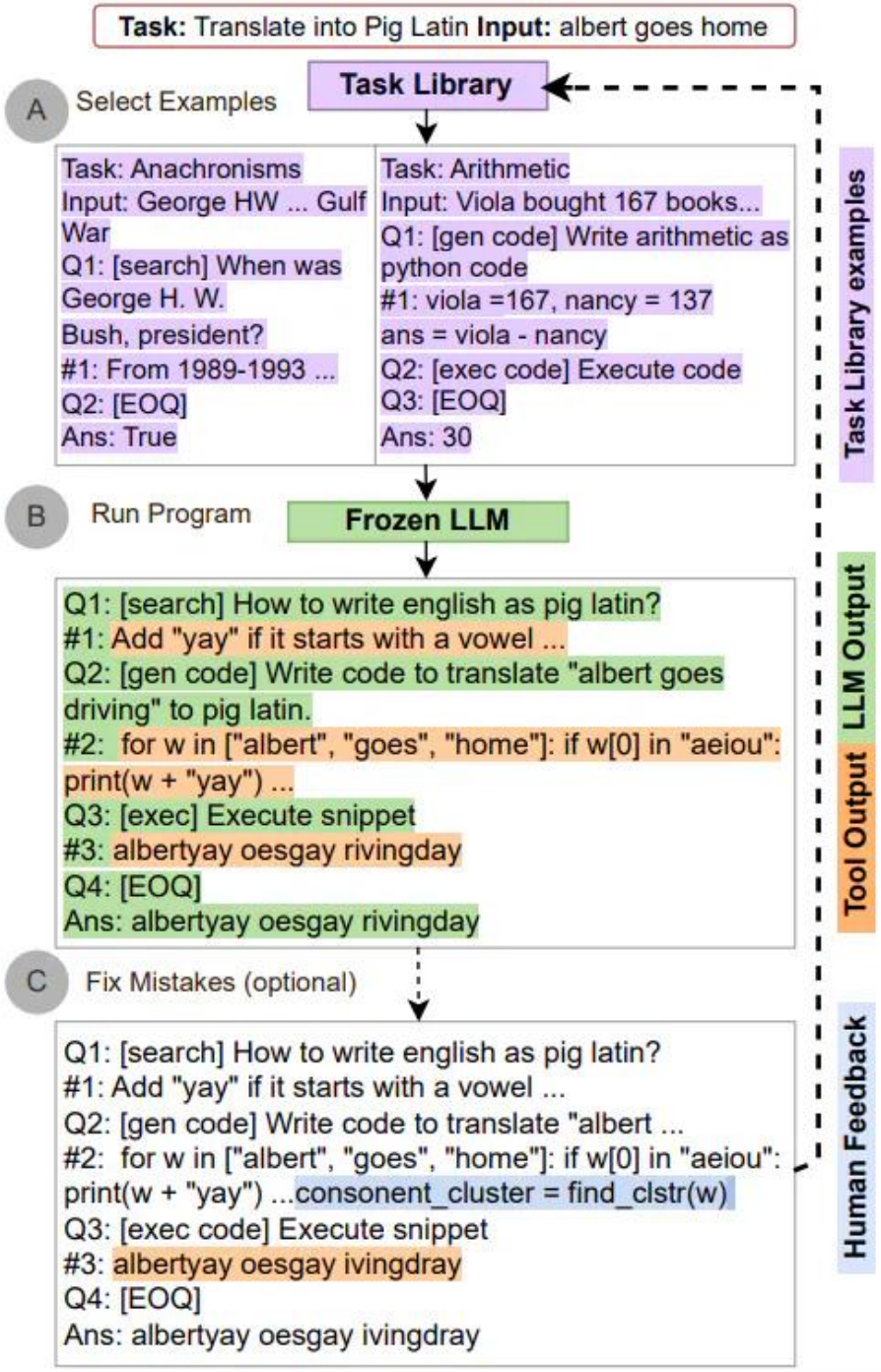
Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.T., Rocktäschel, T. and Riedel, S., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, pp.9459-9474.

Automatic Reasoning and Tool-use (ART)

- given a new task, ART select demonstrations of multi-step reasoning and tool use from a task library
- at test time, ART pauses generation whenever external tools are called, and integrates their output before resuming generation
- ART improves over few-shot prompting and automatic CoT on several unseen tasks exceeds performance of hand-crafted CoT prompts when human feedback is incorporated

Paranjape, B., Lundberg, S., Singh, S., Hajishirzi, H., Zettlemoyer, L. and Ribeiro, M.T., 2023. ART: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*.

ART example

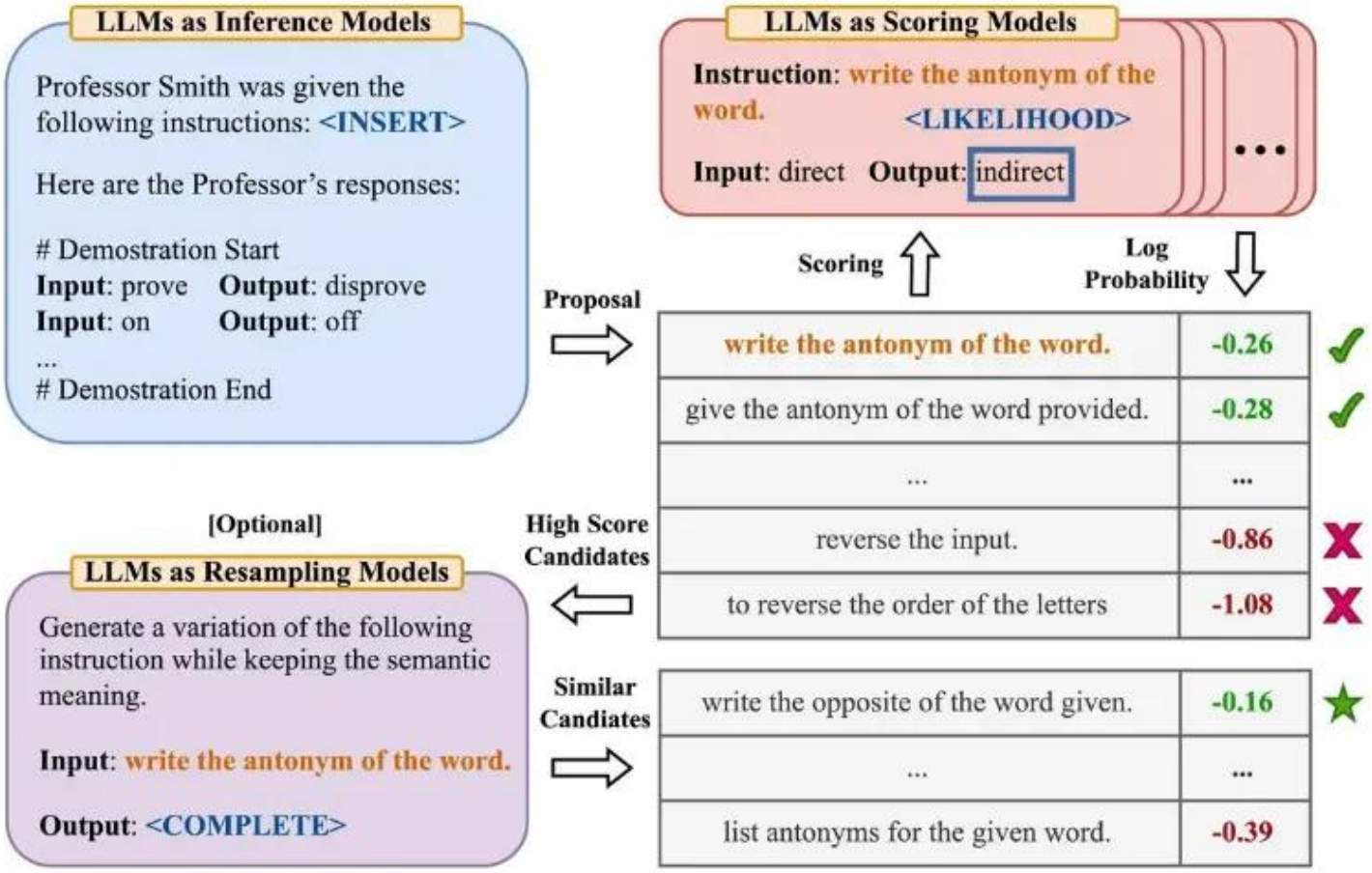

Task: Translate into Pig Latin Input: albert goes home
Task Library
A
Select Examples
Task: Anachronisms
Input: George HW ... Gulf War
Q1: [search] When was George H. W. Bush, president?
#1: From 1989-1993 ...
Q2: [EOQ]
Ans: True
Task: Arithmetic
Input: Viola bought 167 books...
Q1: [gen code] Write arithmetic as python code
#1: viola =167, nancy = 137
ans = viola - nancy
Q2: [exec code] Execute code
Q3: [EOQ]
Ans: 30
Task Library examples
B
Run Program
Frozen LLM
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
LLM Output
Tool Output
C
Fix Mistakes (optional)
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
Human Feedback

Automatic Prompt Engineer (APE)

- A framework for automatic instruction generation and selection.
- The instruction generation problem is framed as natural language synthesis addressed as a black-box optimization problem using LLMs to generate and search over candidate solutions.
- The first step involves a large language model (as an inference model) that is given output demonstrations to generate instruction candidates for a task.
- These candidate solutions will guide the search procedure.
- The instructions are executed using a target model, and then the most appropriate instruction is selected based on computed evaluation scores.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022, November). Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

APE schema


LLMs as Inference Models
Professor Smith was given the following instructions: <INSERT>
Here are the Professor's responses:
Demostration Start
Input: prove Output: disprove
Input: on Output: off
...
Demostration End
LLMs as Scoring Models
Instruction: write the antonym of the word. <LIKELIHOOD>
Input: direct Output: indirect
Scoring
Log Probability
Proposal
write the antonym of the word. -0.26
give the antonym of the word provided. -0.28
... ...
High Score Candidates
reverse the input. -0.86
to reverse the order of the letters -1.08
[Optional]
LLMs as Resampling Models
Generate a variation of the following instruction while keeping the semantic meaning.
Input: write the antonym of the word.
Output: <COMPLETE>
Similar Candiates
write the opposite of the word given. -0.16
... ...
list antonyms for the given word. -0.39

PAL (Program-Aided Language Models)

- The method uses LLMs to read natural language problems and generate programs as the intermediate reasoning steps.
- PAL differs from CoT prompting in that instead of using free-form text to obtain solution it offloads the solution step to a programmatic runtime such as a Python interpreter

Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., ... & Neubig, G. (2023, July). Pal: Program-aided language models. In *International Conference on Machine Learning* (pp. 10764-10799). PMLR. https://arxiv.org/abs/2211.10435

PAL example

Chain-of-Thought (Wei et al., 2022)

Input

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 tennis balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves of bread did they have left?

Model Output

A: The bakers started with 200 loaves. They sold 93 in the morning and 39 in the afternoon. So they sold 93 + 39 = 132 loaves. The grocery store returned 6 loaves. So they had 200 - 132 - 6 = 62 loaves left.
The answer is 62.

Program-aided Language models (this work)

Input

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 tennis balls.
`tennis_balls = 5`
2 cans of 3 tennis balls each is
`bought_balls = 2 * 3`
tennis balls. The answer is
`answer = tennis_balls + bought_balls`

Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves of bread did they have left?

Model Output

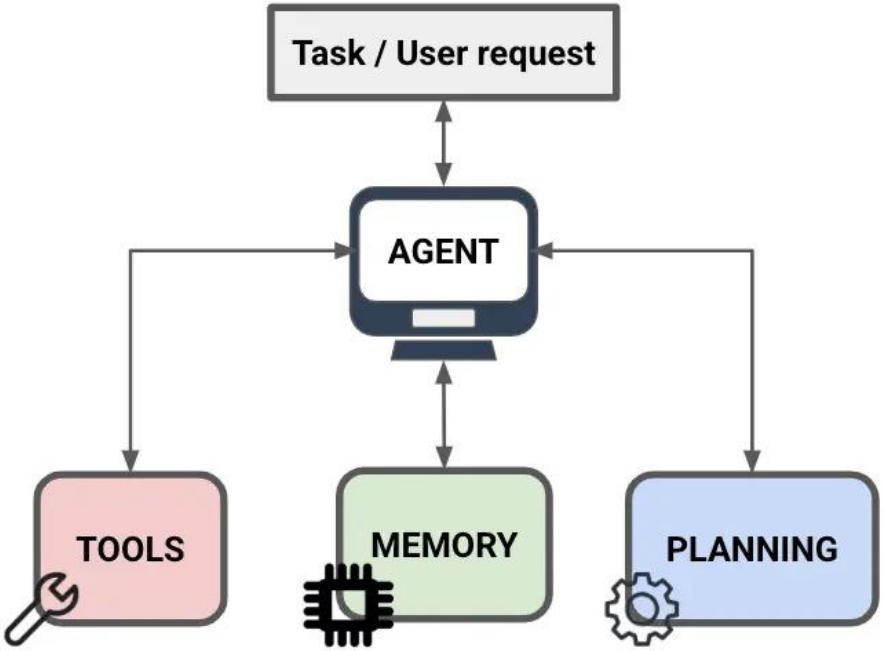
A: The bakers started with 200 loaves
`loaves_baked = 200`
They sold 93 in the morning and 39 in the afternoon
`loaves_sold_morning = 93`
`loaves_sold_afternoon = 39`
The grocery store returned 6 loaves.
`loaves_returned = 6`
The answer is

```
answer = loaves_baked - loaves_sold_morning
  - loaves_sold_afternoon + loaves_returned
```

```
>>> print(answer)
74
```

LLM Agents

- A relatively new approach utilizing LLMs for various complex tasks



Additional capabilities of LLM Agents

- Planning and reflection: AI agents can analyze a problem, break it down into steps, and adjust their approach based on new information.
- Tool access: They can interact with external tools and resources, such as databases, APIs, and software applications, to gather information and execute actions.
- Memory: AI agents can store and retrieve information, allowing them to learn from past experiences and make more informed decisions.

Agent applications

- **Recommendation systems:** Personalizing suggestions for products, services, or content.
- **Customer support systems:** Handling inquiries, resolving issues, and providing assistance.
- **Research:** Conducting in-depth investigations across various domains, such as legal, finance, and health.
- **E-commerce applications:** Facilitating online shopping experiences, managing orders, and providing personalized recommendations.
- **Booking:** Assisting with travel arrangements and event planning.
- **Reporting:** Analyzing vast amounts of data and generating comprehensive reports.
- **Financial analysis:** Analyzing market trends, assess financial data, and generate reports with unprecedented speed and accuracy.