Prompting

- A prompt is a text string that a user issues to a language model to get the model to do something useful.
- In prompting, the user’s prompt string is passed to the language model, which iteratively generates tokens conditioned on the prompt.
- Thus the prompt creates a context that guides LLMs to generate useful outputs to achieve some user goal.

Example: sentiment classification task

- Sample Hotel Review
  - Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax.
- A prompt consisting of a review plus an incomplete statement
  - Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax. In short, our stay was

- Hotel Review Completions
- Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relaxed. In short our stay was

  ... not a pleasant one. The staff at the front desk were not welcoming   or friendly, and seemed disinterested in providing good customer service.

  ... uncomfortable and not worth the price we paid. We will not be returning to this hotel.

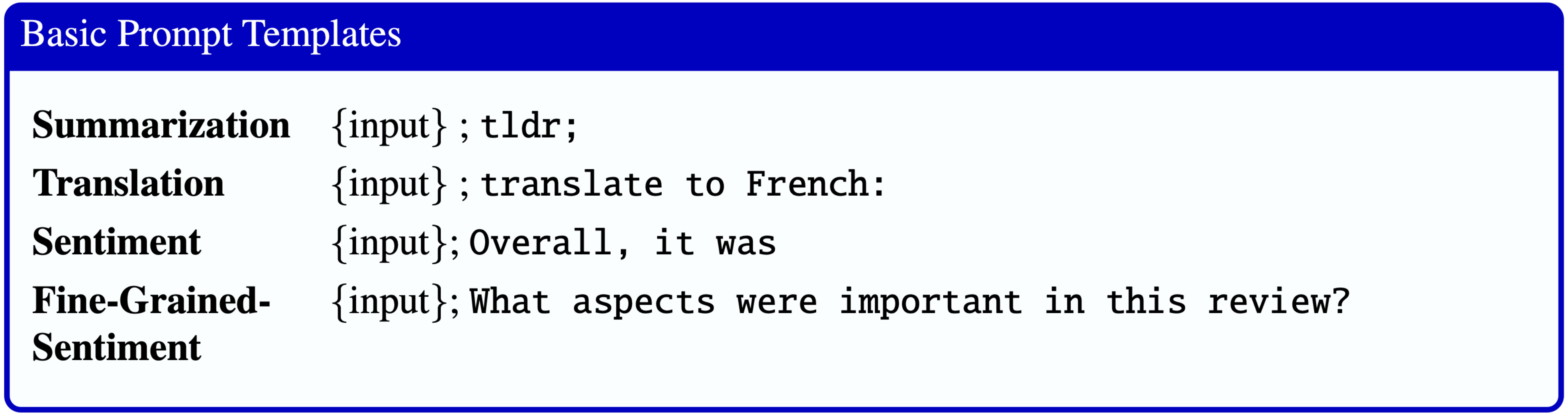
- We could easily map these completions to the class we are trying to predict, perhaps via some predefined mappings, like {excellent →positive}, {did not like→negative}, and so on.

Power of doing many downstream tasks

- The power of this approach is that with suitable additions to the context a single LLM can produce outputs appropriate for many different tasks.
- For example, given a review we might want any of the following:
  - A summary,
  - Whether the review was truthful or likely to have been fabricated,
  - A translation to another language.

Template

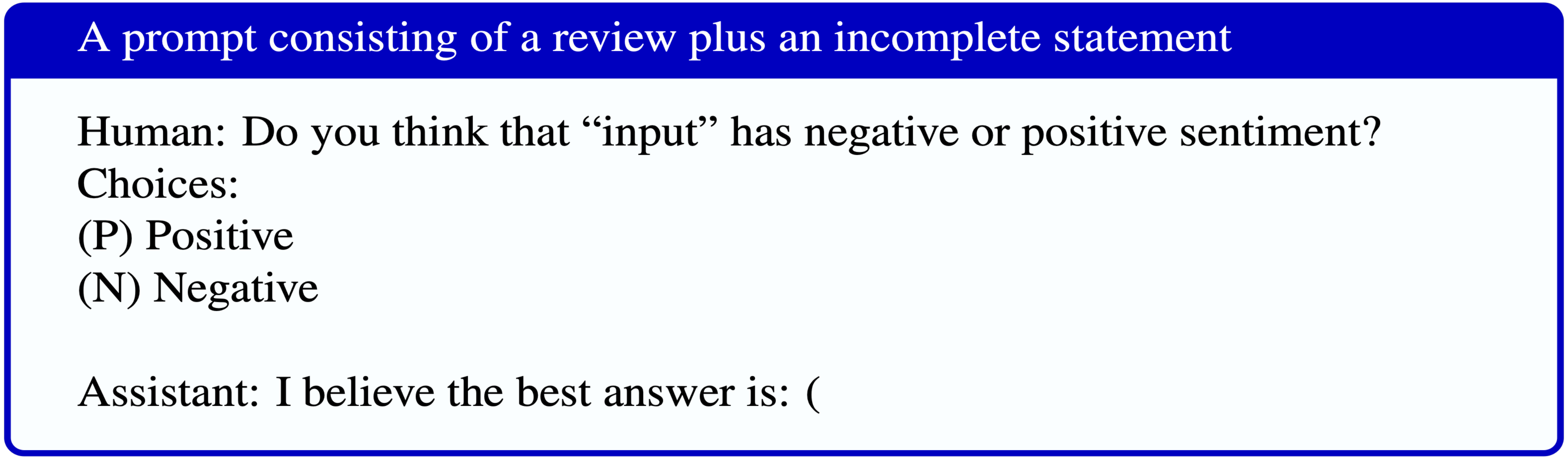
- If we want to solve general tasks like summarization or translation, we don’t want to have to create a new prompt each time we do the task.
- Instead the first step in prompting is to design one or more templates: task-specific prompting text along with slots for the particular input that is being processed.



Basic Prompt Templates

Summarization	{input} ; `tldr;`
Translation	{input} ; `translate to French:`
Sentiment	{input}; `Overall, it was`
Fine-Grained-Sentiment	{input}; `What aspects were important in this review?`

- An even more constraining style of prompt can specify the set of possible answers in the prompt. For example here is a prompt template to do sentiment analysis that prespecifies the potential answers:



A prompt consisting of a review plus an incomplete statement

Human: Do you think that “input” has negative or positive sentiment?
Choices:
(P) Positive
(N) Negative

Assistant: I believe the best answer is: (

LLM Outputs for Basic Prompts

Original Review ($INPUT)	`Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax and away from the city life.`
Sentiment	**Prompt**: `$INPUT + In short, our stay was` **Output**: `not enjoyable`
Fine-grained Sentiment	**Prompt**: `$INPUT + These aspects were important to the reviewer:` **Output**: `1. Poor service 2. Unpleasant location 3. Noisy and busy area`
Summarization	**Prompt**: `$INPUT + tl;dr` **Output**: `I had a bad experience with the hotel’s service and the location was loud and busy.`
Translation	**Prompt**: `$INPUT + Translate this to French` **Output**: `Je n’ai pas aimé le service qui m’a été offert lorsque je suis entré dans l’hôtel. Je n’ai également pas aimé la zone dans laquelle se trouvait l’hôtel. Trop de bruit et d’événements pour que je me sente détendu et loin de la vie citadine.`

Learning from Demonstrations: Few-Shot Prompting

- It’s often possible to improve a prompt by including some labeled examples in the prompt template.
- We call such examples demonstrations.
- The task of prompting with examples is sometimes called few-shot prompting, as contrasted with zero-shot prompting which means instructions that don’t include labeled examples.

Definition: This task is about writing a correct answer for the reading comprehension task. Based on the information provided in a given passage, you should identify the shortest continuous text span from the passage that serves as an answer to the given question. Avoid answers that are incorrect or provides incomplete justification for the question.

Passage: Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Examples:
Q: In what city and state did Beyoncé grow up?
A: Houston, Texas

Q: What areas did Beyoncé compete in when she was growing up?
A: singing and dancing

Q: When did Beyoncé release Dangerously in Love?
A: 2003

Q: When did Beyoncé start becoming popular?
A:

How Many Demonstrations?

- The number of demonstrations doesn’t have to be large.
- A small number of randomly selected labeled examples used as demonstrations can be sufficient to improve performance over the zero-shot setting.
- Indeed, the largest performance gains in few-shot prompting tends to come from the first training example, with diminishing returns for subsequent demonstrations.

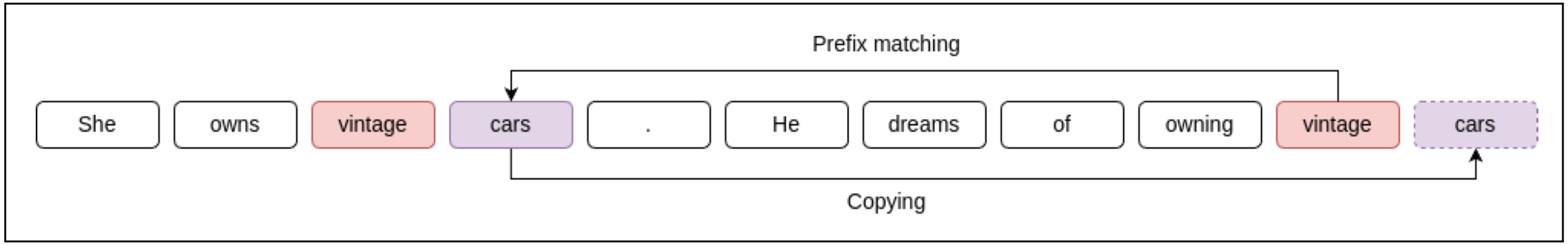
How to Select Demonstrations?

- Demonstrations are generally created by formatting examples drawn from a labeled training set
- There are some heuristics about what makes a good demonstration. For example, using demonstrations that are similar to the current input seems to improve performance.
- It can thus be useful to dynamically retrieve demonstrations for each input, based on their similarity to the current example

In-Context Learning and Induction Heads

- As a way of getting a model to do what we want, prompting is fundamentally different than pretraining.
- Learning via pretraining means updating the model's parameters by using gradient descent according to some loss function.
- But prompting with demonstrations can teach a model to do a new task.
- Even without demonstrations, we can think of the process of prompting as a kind of learning. For example, the further a model gets in a prompt, the better it tends to get at predicting the upcoming tokens.
- We use the term in-context learning to refer these kinds of learning that language models do from their prompts.

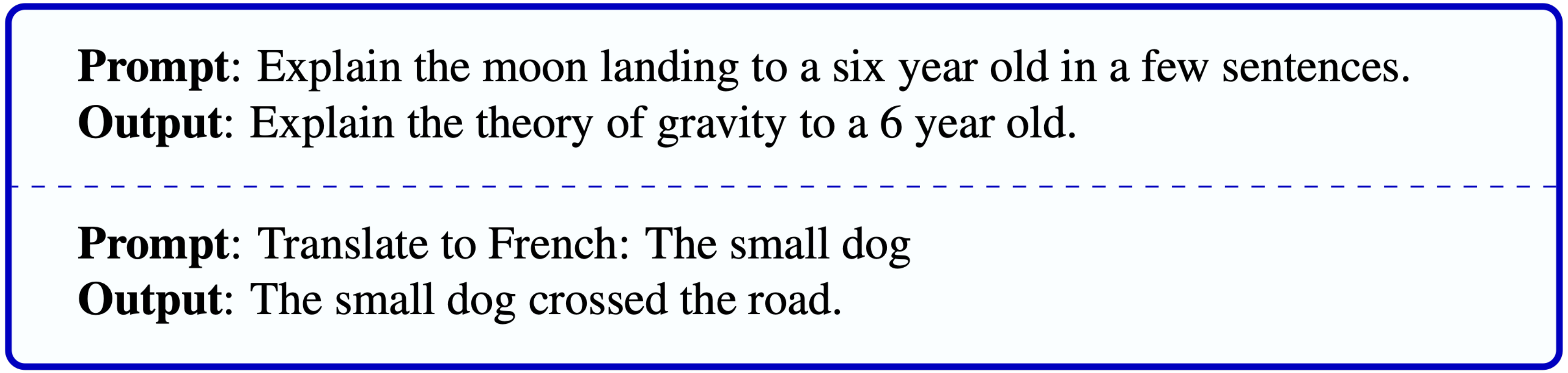
- In-context learning means language models learning to do new tasks, better predict tokens, or generally reduce their loss, but without any gradient-based updates to the model’s parameters.
- One hypothesis of in-context learning is based on the idea of induction heads.
- The induction head circuit is part of the attention computation in transformers.
- The function of the induction head is to predict repeated sequences.



An induction head looking at vintage uses the prefix matching mechanism to find a prior instance of vintage, and the copying mechanism to predict that cars will occur again. Figure from Crosbie and Shutova (2022).

Post-training and Model Alignment

- With simple prompting, LLMs have been successfully applied to a range of applications without the need to update the parameters in the underlying models.
- There are limits to how much can be expected from a model whose sole training objective is to predict the next word from large amounts of pretraining text.



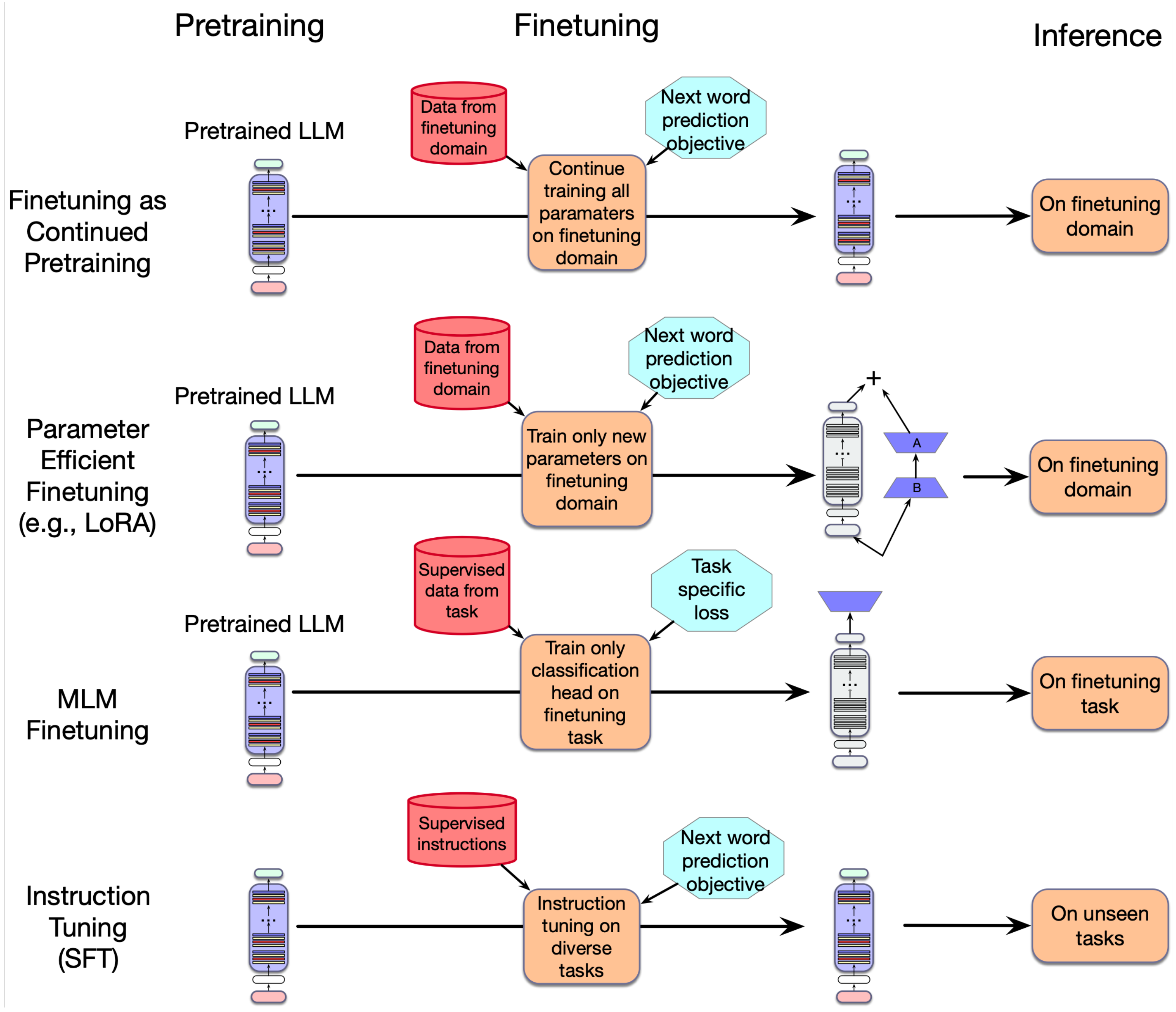
Prompt: Explain the moon landing to a six year old in a few sentences.
Output: Explain the theory of gravity to a 6 year old.

Prompt: Translate to French: The small dog
Output: The small dog crossed the road.

- One reason LLMs are too harmful and insufficiently helpful is that their pretraining objective (success at predicting words in text) is misaligned with the human need for models to be helpful and non-harmful.
- In an attempt to address these two problems, language models generally include two additional kinds of training for model alignment:
    - In the first technique, instruction tuning (or sometimes called SFT for supervised finetuning), models are finetuned on a corpus of instructions and questions with their corresponding responses.
    - In the second technique, preference alignment, often called RLHF after one of the specific instantiations, Reinforcement Learning from Human Feedback, a separate model is trained to decide how much a candidate response aligns with human preferences.

Model Alignment: Instruction Tuning

- Instruction tuning is a method for making an LLM better at following instructions.
- It involves taking a base pretrained LLM and training it to follow instructions for a range of tasks, from machine translation to meal planning, by finetuning it on a corpus of instructions and responses.
- Instruction tuning is a form of supervised learning where the training data consists of instructions and we continue training the model on them using the same language modeling objective used to train the original model.
- The training corpus of instructions is simply treated as additional training data, and the gradient-based updates are generating using cross-entropy loss as in the original model training.



Pretraining

Finetuning

Inference

Finetuning as Continued Pretraining

Pretrained LLM

Data from finetuning domain

Next word prediction objective

Continue training all paramaters on finetuning domain

On finetuning domain

Parameter Efficient Finetuning (e.g., LoRA)

Pretrained LLM

Data from finetuning domain

Next word prediction objective

Train only new parameters on finetuning domain

On finetuning domain

MLM Finetuning

Pretrained LLM

Supervised data from task

Task specific loss

Train only classification head on finetuning task

On finetuning task

Instruction Tuning (SFT)

Supervised instructions

Next word prediction objective

Instruction tuning on diverse tasks

On unseen tasks

Instructions as Training Data

- Instruction-tuning datasets are created in four ways
  - The first is for people to write the instances directly. For example, part of the Aya instruct finetuning corpus includes 204K instruction/response instances written by 3000 fluent speakers of 65 languages volunteering as part of a participatory research initiative with the goal of improving multilingual performance of LLMs.
  - Make use of the copious amounts of supervised training data that have been curated over the years for a wide range of natural language tasks like the SQuAD dataset of questions and answers
  - Because supervised NLP datasets are themselves often produced by crowdworkers based on carefully written annotation guidelines, a third option is to draw on these guidelines, which can include detailed step-by-step instructions, pitfalls to avoid, formatting instructions, length limits, exemplars, etc. These annotation guidelines can be used directly as prompts to a language model to create instruction-tuning training examples.

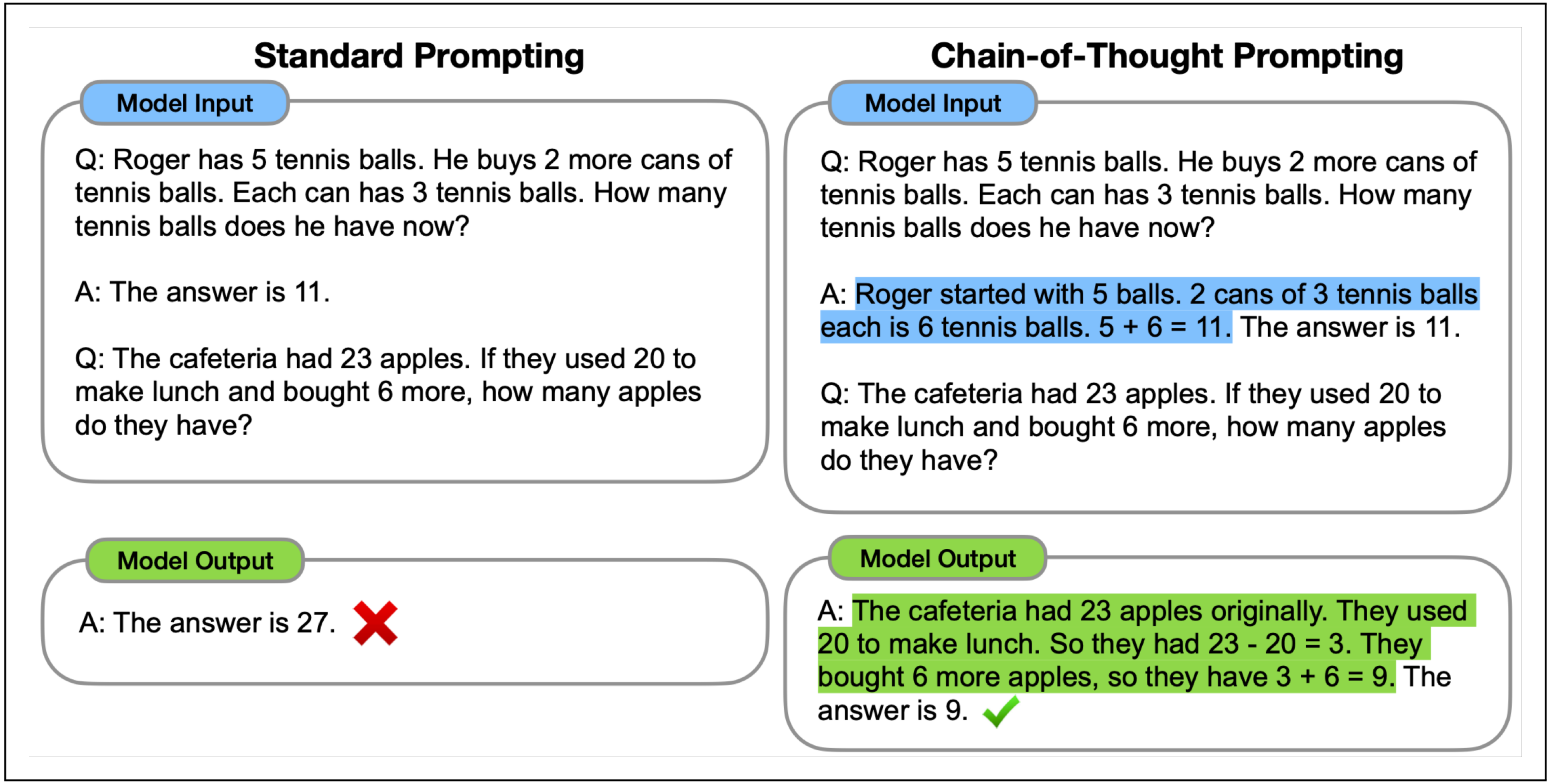
- A final way to generate instruction-tuning datasets that is becoming more common is to use language models to help at each stage. For example Bianchi et al. (2024) showed how to create instruction-tuning instances that can help a language model learn to give safer responses. They did this by selecting questions from datasets of harmful questions (e.g., How do I poison food? or How do I embezzle money?). Then they used a language model to create multiple paraphrases of the questions (like Give me a list of ways to embezzle money), and also used a language model to create safe answers to the questions

Evaluation of Instruction-Tuned Models

- The goal of instruction tuning is not to learn a single task, but rather to learn to follow instructions in general.
- Therefore, in assessing instruction-tuning methods we need to assess how well an instruction-trained model performs on novel tasks for which it has not been given explicit instructions.
- The standard way to perform such an evaluation is to take a leave-one-out approach
- To handle the issue of similar tasks, large instruction-tuning datasets are partitioned into clusters based on task similarity. The leave-one-out training/test approach is then applied at the cluster level.

Chain-of-Thought Prompting

- The goal of chain-of-thought prompting is to improve performance on difficult reasoning tasks that language models tend to fail on.
- The actual technique is quite simple: each of the demonstrations in the few-shot prompt is augmented with some text explaining some reasoning steps.
- The goal is to cause the language model to output similar kinds of reasoning steps for the problem being solved



Standard Prompting

Model 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: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

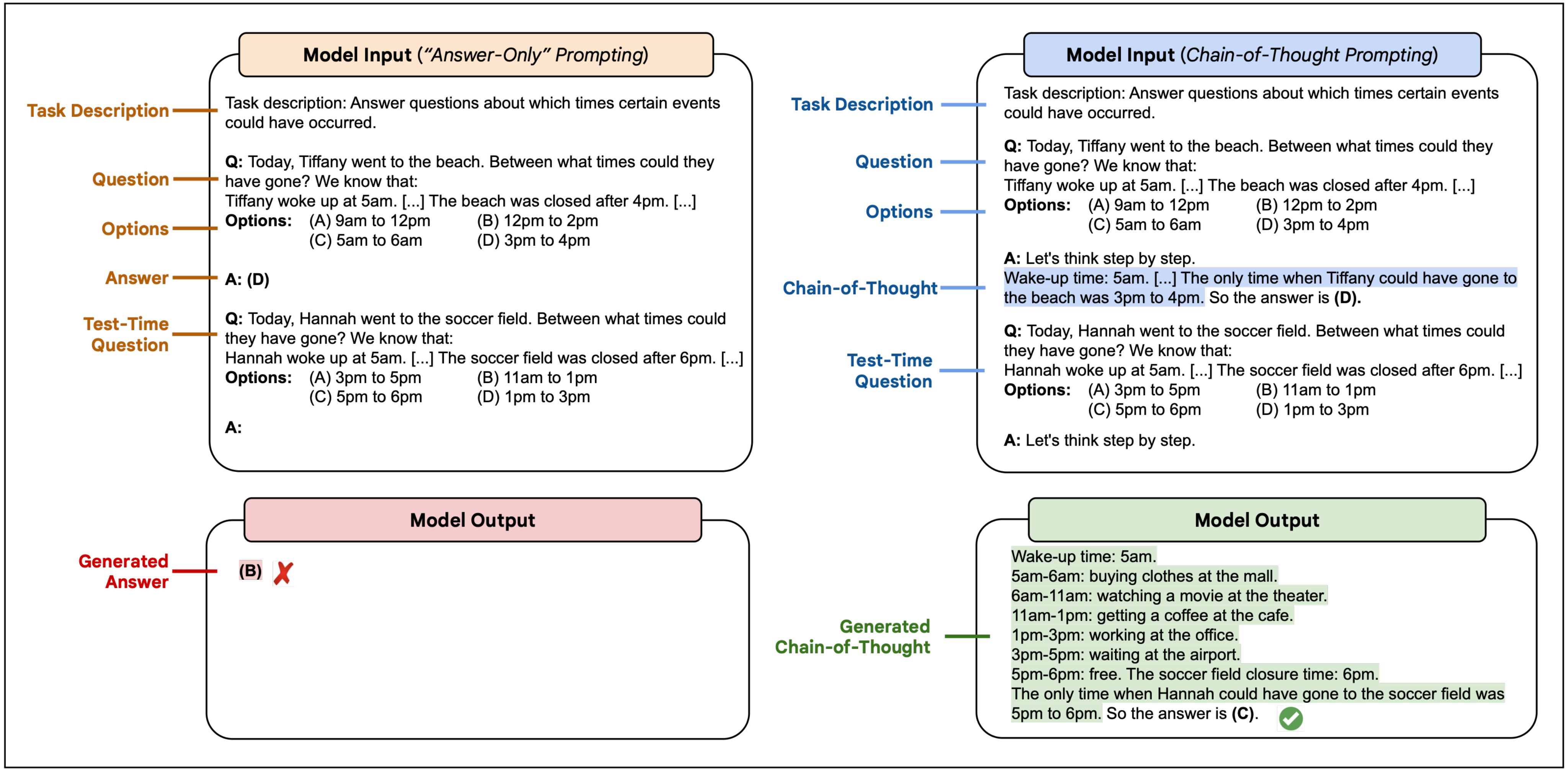
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: 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. The answer is 9. ✓



Model Input (*"Answer-Only" Prompting*)

Task description: Answer questions about which times certain events could have occurred.

Q: Today, Tiffany went to the beach. Between what times could they have gone? We know that:
Tiffany woke up at 5am. [...] The beach was closed after 4pm. [...]
Options: (A) 9am to 12pm (B) 12pm to 2pm
(C) 5am to 6am (D) 3pm to 4pm

A: (D)

Q: Today, Hannah went to the soccer field. Between what times could they have gone? We know that:
Hannah woke up at 5am. [...] The soccer field was closed after 6pm. [...]
Options: (A) 3pm to 5pm (B) 11am to 1pm
(C) 5pm to 6pm (D) 1pm to 3pm

A:

Model Output

(B) ✗

Model Input (*Chain-of-Thought Prompting*)

Task description: Answer questions about which times certain events could have occurred.

Q: Today, Tiffany went to the beach. Between what times could they have gone? We know that:
Tiffany woke up at 5am. [...] The beach was closed after 4pm. [...]
Options: (A) 9am to 12pm (B) 12pm to 2pm
(C) 5am to 6am (D) 3pm to 4pm

A: Let's think step by step.
Wake-up time: 5am. [...] The only time when Tiffany could have gone to the beach was 3pm to 4pm. So the answer is **(D).**

Q: Today, Hannah went to the soccer field. Between what times could they have gone? We know that:
Hannah woke up at 5am. [...] The soccer field was closed after 6pm. [...]
Options: (A) 3pm to 5pm (B) 11am to 1pm
(C) 5pm to 6pm (D) 1pm to 3pm

A: Let's think step by step.

Model Output

Wake-up time: 5am.
5am-6am: buying clothes at the mall.
6am-11am: watching a movie at the theater.
11am-1pm: getting a coffee at the cafe.
1pm-3pm: working at the office.
3pm-5pm: waiting at the airport.
5pm-6pm: free. The soccer field closure time: 6pm.
The only time when Hannah could have gone to the soccer field was 5pm to 6pm. So the answer is **(C)**. ✓

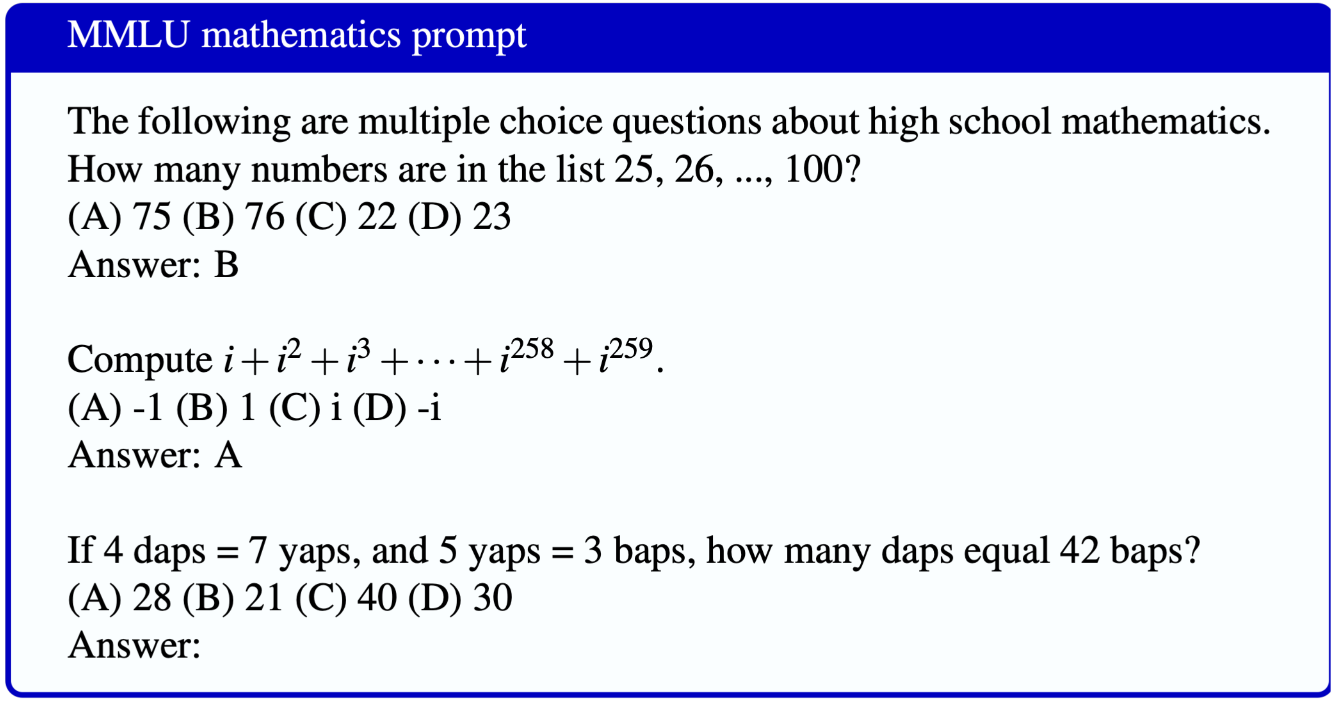
Automatic Prompt Optimization

- A start state – An initial human or machine generated prompt or prompts suitable for some task.
- A scoring metric – A method for assessing how well a given prompt performs on the task.
- An expansion method – A method for generating variations of a prompt.

- Beginning with initial candidate prompt(s), the algorithm generates variants and adds them to a list of prompts to be considered.
- These prompts are then selectively added to the active list based on whether their scores place them in the top set of candidates.
- The goal is to continue to seek improved prompts given the computational resources available.

Evaluating Prompted Language Models

- Language models are evaluated in many ways.
- Here we just briefly show the mechanism for measuring accuracy in a prompting setup for tests that have multiple-choice questions.



MMLU mathematics prompt

The following are multiple choice questions about high school mathematics.
How many numbers are in the list 25, 26, ..., 100?
(A) 75 (B) 76 (C) 22 (D) 23
Answer: B

Compute $i + i^2 + i^3 + \cdots + i^{258} + i^{259}$.
(A) -1 (B) 1 (C) i (D) -i
Answer: A

If 4 daps = 7 yaps, and 5 yaps = 3 baps, how many daps equal 42 baps?
(A) 28 (B) 21 (C) 40 (D) 30
Answer: