Building Large Language Models

COMP7607 — Lecture 8

Lingpeng Kong

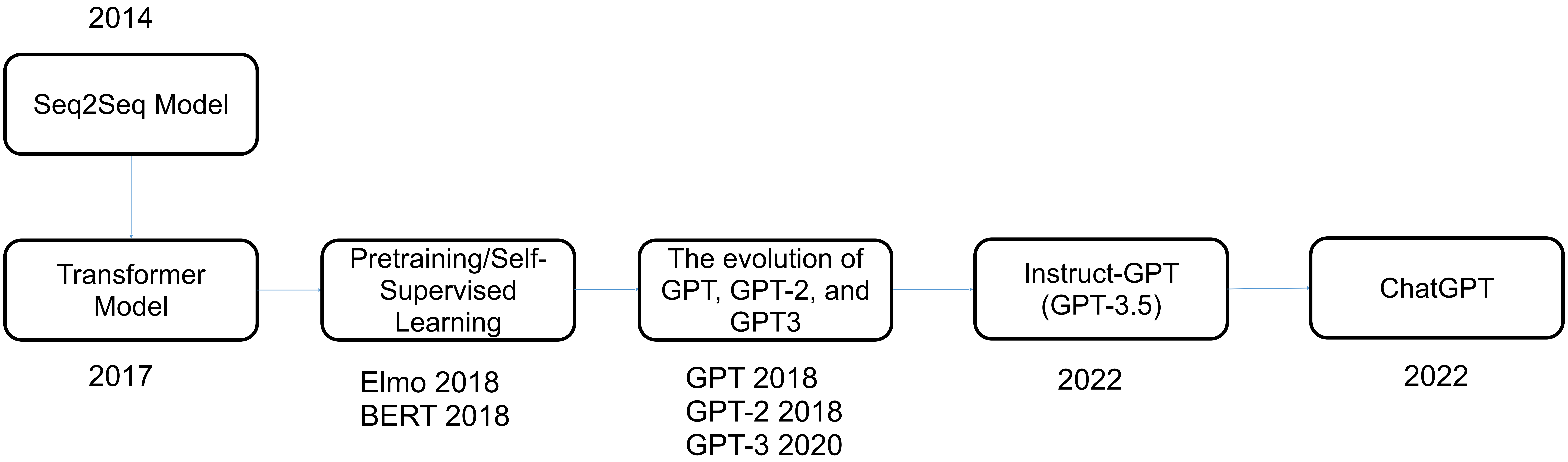
Department of Computer Science, The University of Hong Kong

Most slides from Dr. Qi Liu with special thanks!

I. Overview: What happened during the past few years?

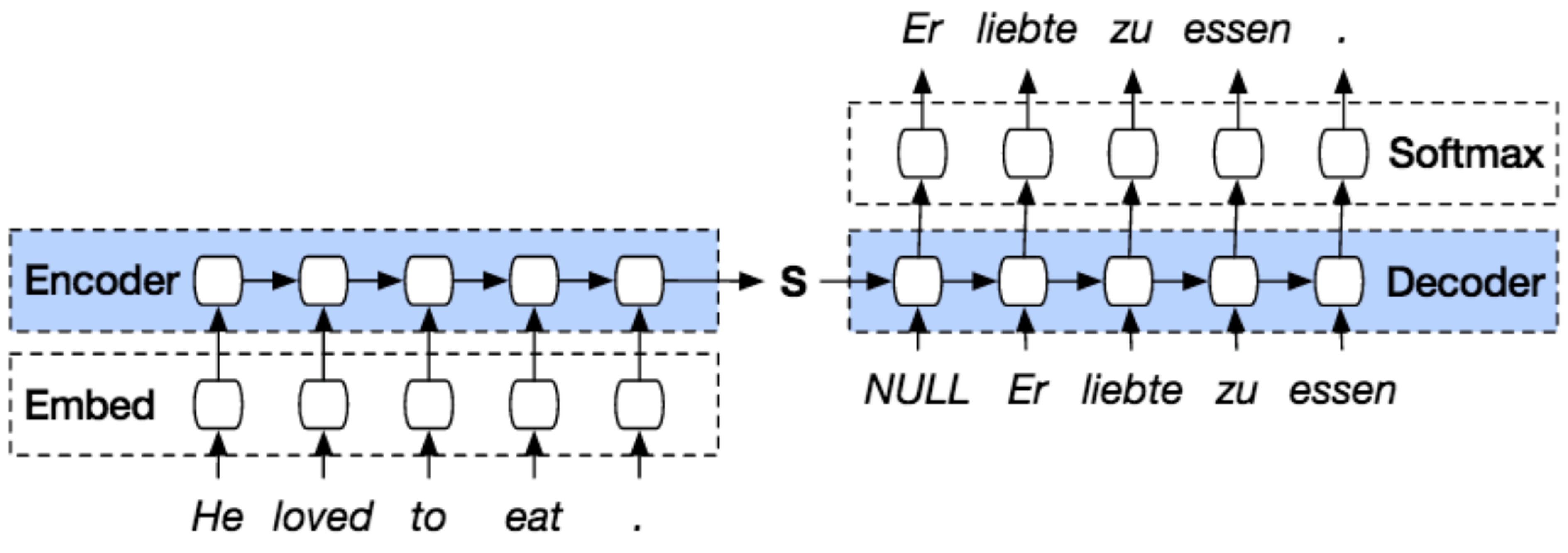
We introduce:

- The idea of sequence2sequence
- The transformer model
- Pretraining the transformer model
- The evolution of GPT, GPT-2, and GPT-3
- Instruct-GPT: aligning language models to follow instructions.
- ChatGPT: Using the technique of Instruct-GPT for chatbot.

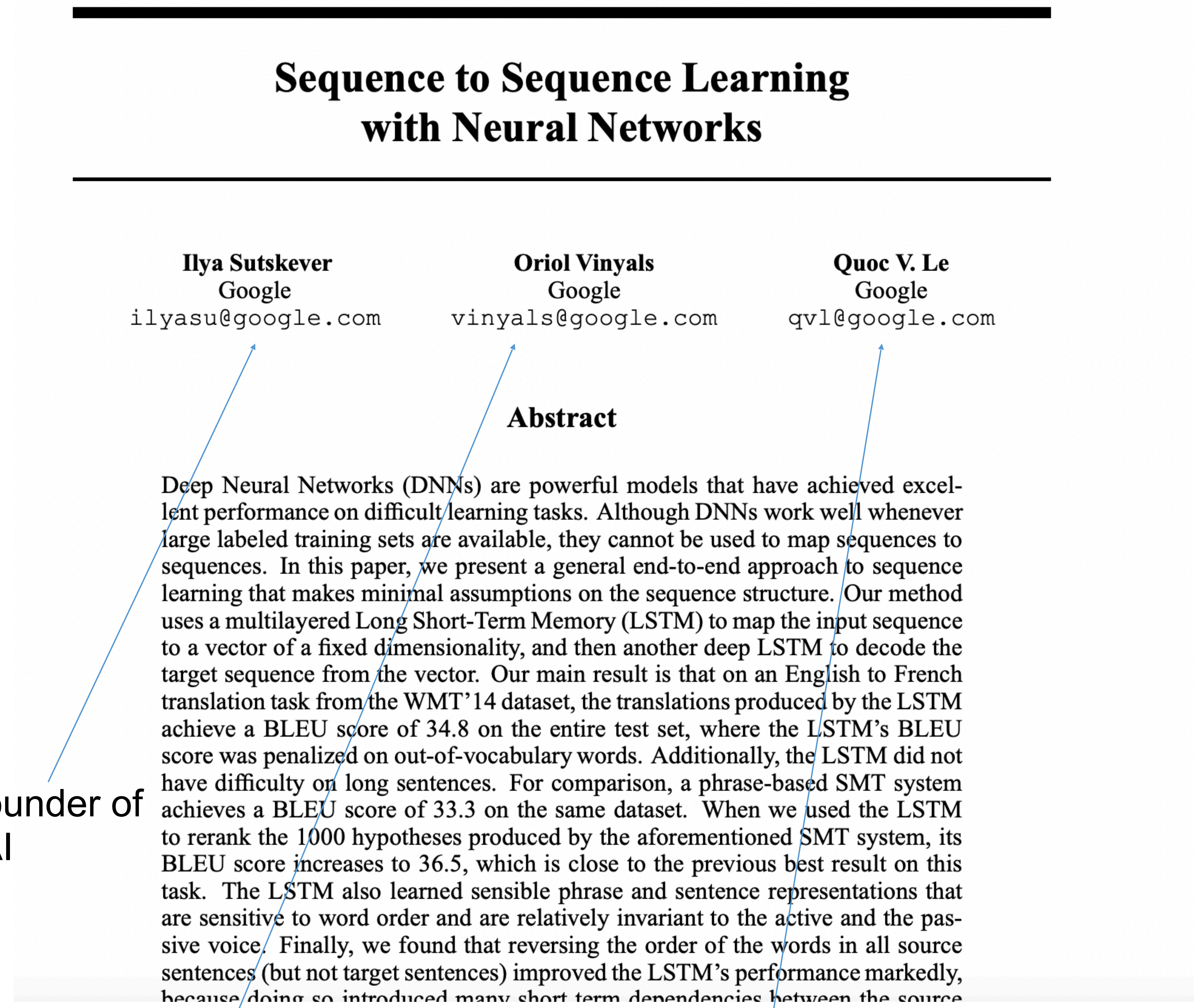


I. Seq2Seq Model: 2014

- Formulate NLP problems into a sequence-to-sequence translation task.
- Machine translation
  - For example, “He loved to eat.” -> 他喜欢吃。
- Seq2Seq uses an LSTM to encode the input sequence and uses another LSTM to decode the output sequence.
- Seq2Seq model is an influential paper that inspires much follow-up research.



Seq2Seq



Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT’14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM’s BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM’s performance markedly, because doing so introduced many short term dependencies between the source

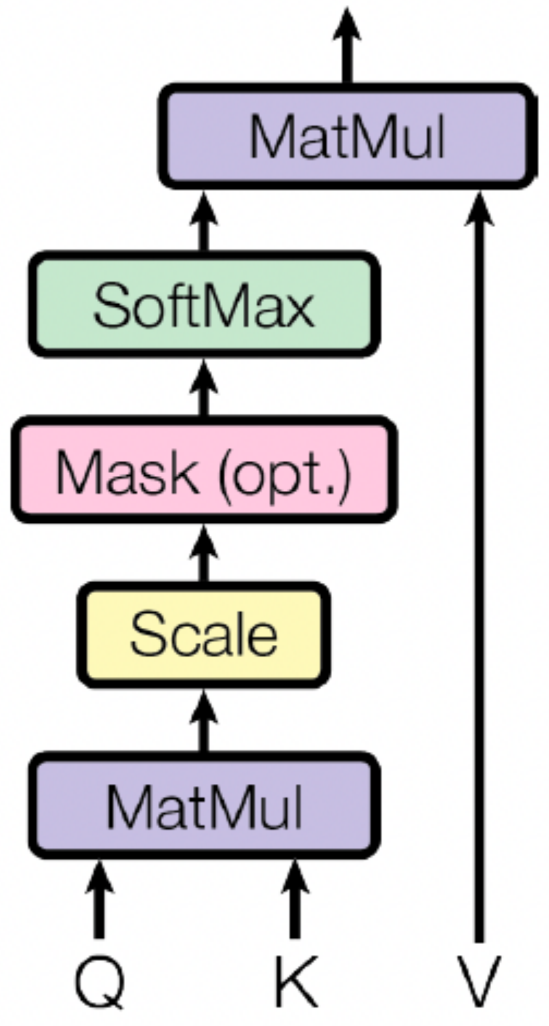
A co-founder of OpenAI

At DeepMind

At Google Brain, inventor of the Chatbot, Meena

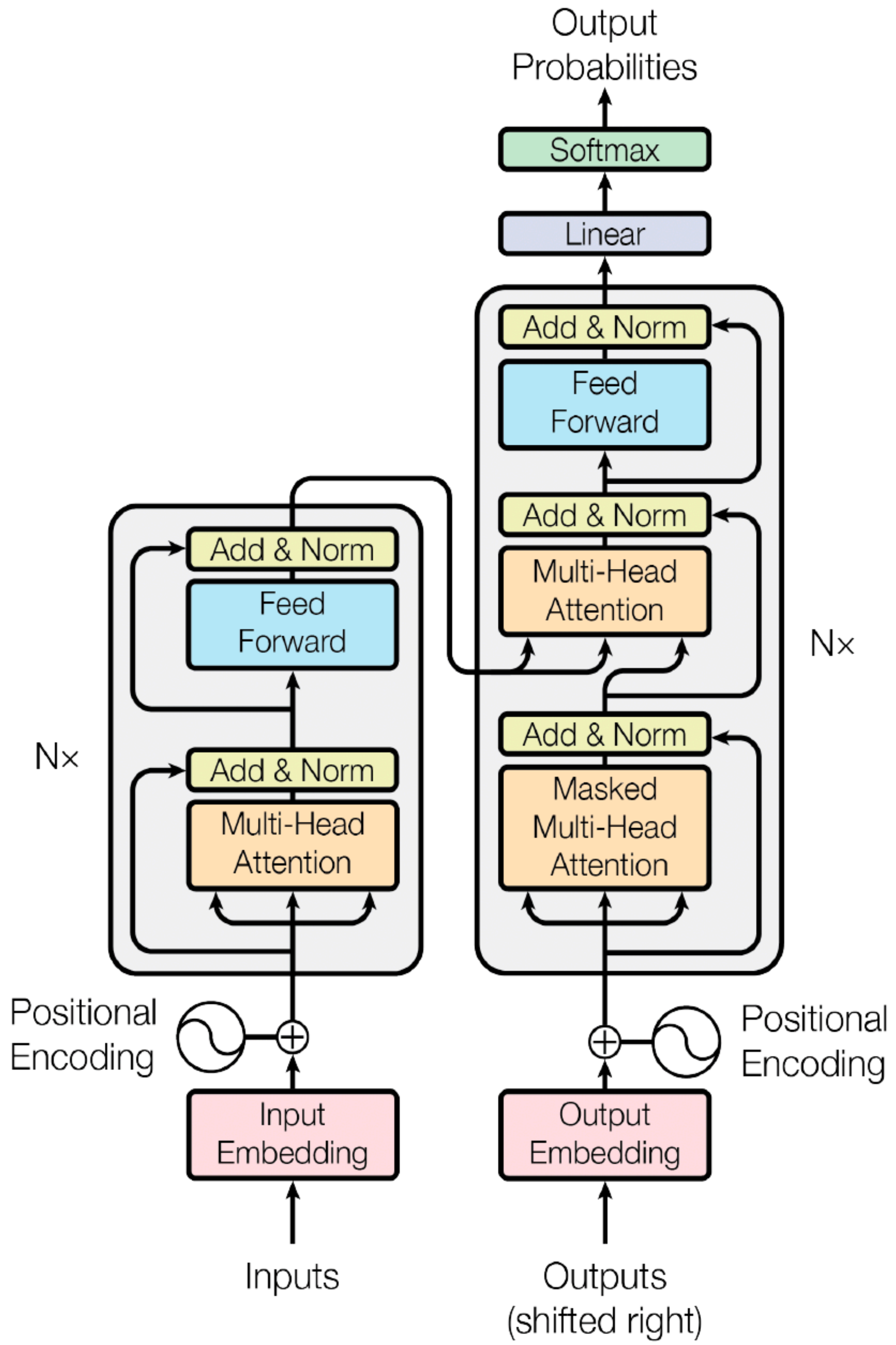
I. Transformer: 2017

- Although LSTM is widely used before 2017, it cannot capture long-term dependency well.
  - It is estimated that LSTM cannot capture the long-term dependency of more than 7-15 tokens.
- **GPT** stands for **G**enerative **P**re-trained **T**ransformer.
- The core idea of **Transformer**: The attention mechanism
  - Given a sequence of n tokens $(t_1, t_2, \ldots, t_n)$, Transformer calculates the attention score between each token pair (t_i, t_j).
  - The calculation is expensive but it is proven to be effective in capturing long-term dependency in practice.



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The self-attention mechanism used by Transformer.



The Transformer - Model Architecture

I. Transformer Authors

Transformer is an influential paper published by Google.
All authors co-founded their own startups or moved to other companies.

Attention Is All You Need

Adept | character.ai | Adept | Inceptive

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

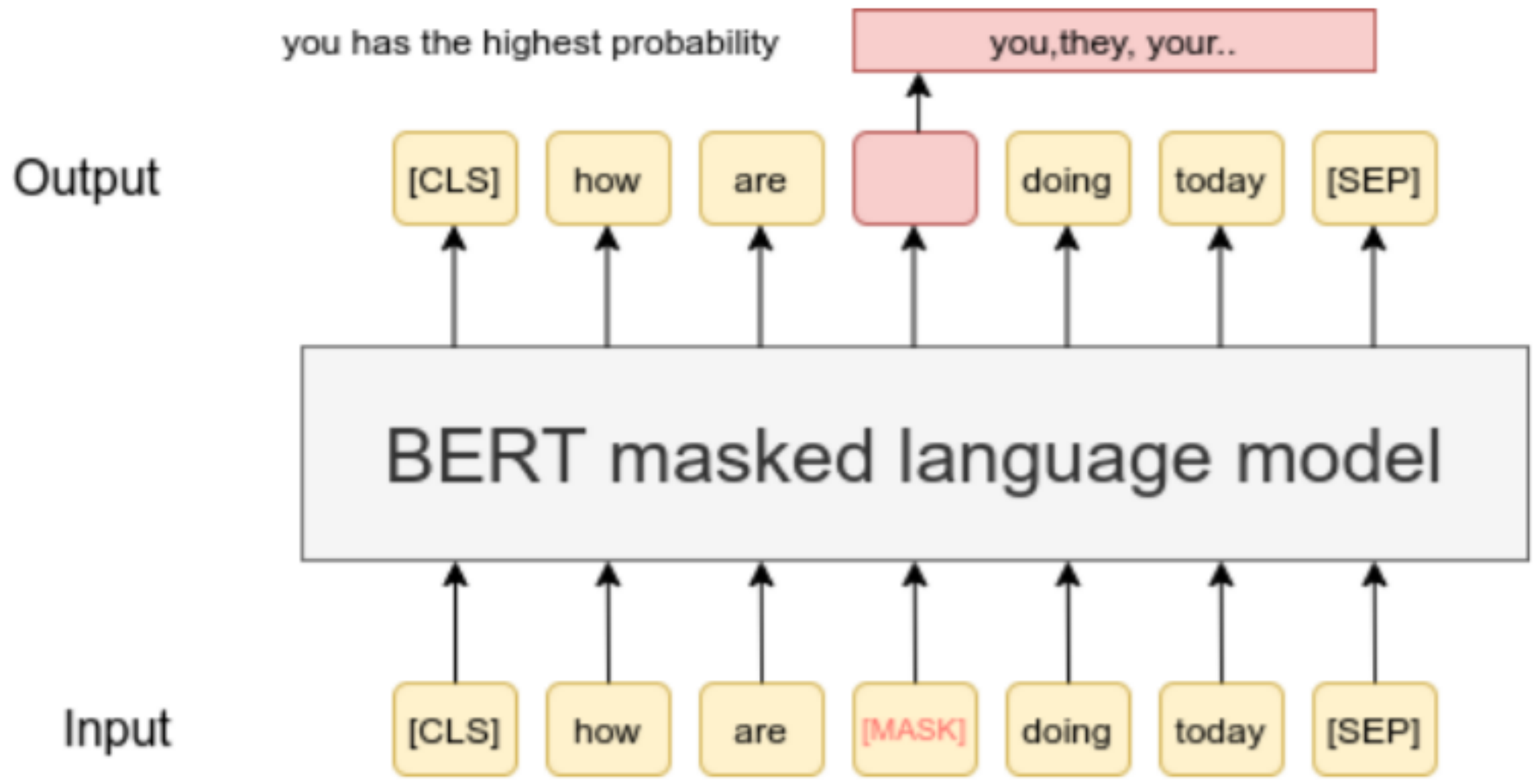
co:here

NEAR INCORPORATED

Sakana AI

I. Pretraining/Self-Supervised Learning

- Transformer is a powerful model itself, but it works best when there are abundant labeled data.
- Pretraining/self-supervised learning is a good solution to mitigate labeled data scarcity.
- Self-supervised learning is a machine learning process where the model trains itself to learn one part of the input from another part of the input.
- When combining Transformer with pretraining/self-supervised learning, an amazing "chemical reaction" happens.



BERT: Some tokens in the input sequence are masked. The pretraining task is to reconstruct the masked tokens

I. Popular Pretraining Objectives in NLP

Language model is used by GPT-3/ChatGPT. This objective is useful for both classification and generation

The masked language model is used by BERT. This objective is mostly useful for classification or natural language understanding tasks.

Objective	Inputs	Targets
LM	[START]	I am happy to join with you today
MLM	I am [MASK] to join with you [MASK]	happy today
NSP	Sent1 [SEP] Next Sent or Sent1 [SEP] Random Sent	Next Sent/Random Sent
SOP	Sent1 [SEP] Sent2 or Sent2 [SEP] Sent1	in order/reversed
Discriminator (o/r)	I am thrilled to study with you today	o o r o r o o o
PLM	happy join with	today am I to you
seq2seq LM	I am happy to	join with you today
Span Mask	I am [MASK] [MASK] [MASK] with you today	happy to join
Text Infilling	I am [MASK] with you today	happy to join
Sent Shuffling	today you am I join with happy to	I am happy to join with you today
TLM	How [MASK] you [SEP] [MASK] vas-tu	are Comment

I. Language Model

GPT Models = Language Model Objective (G) + Pretraining Data (P) + Transformer (T)

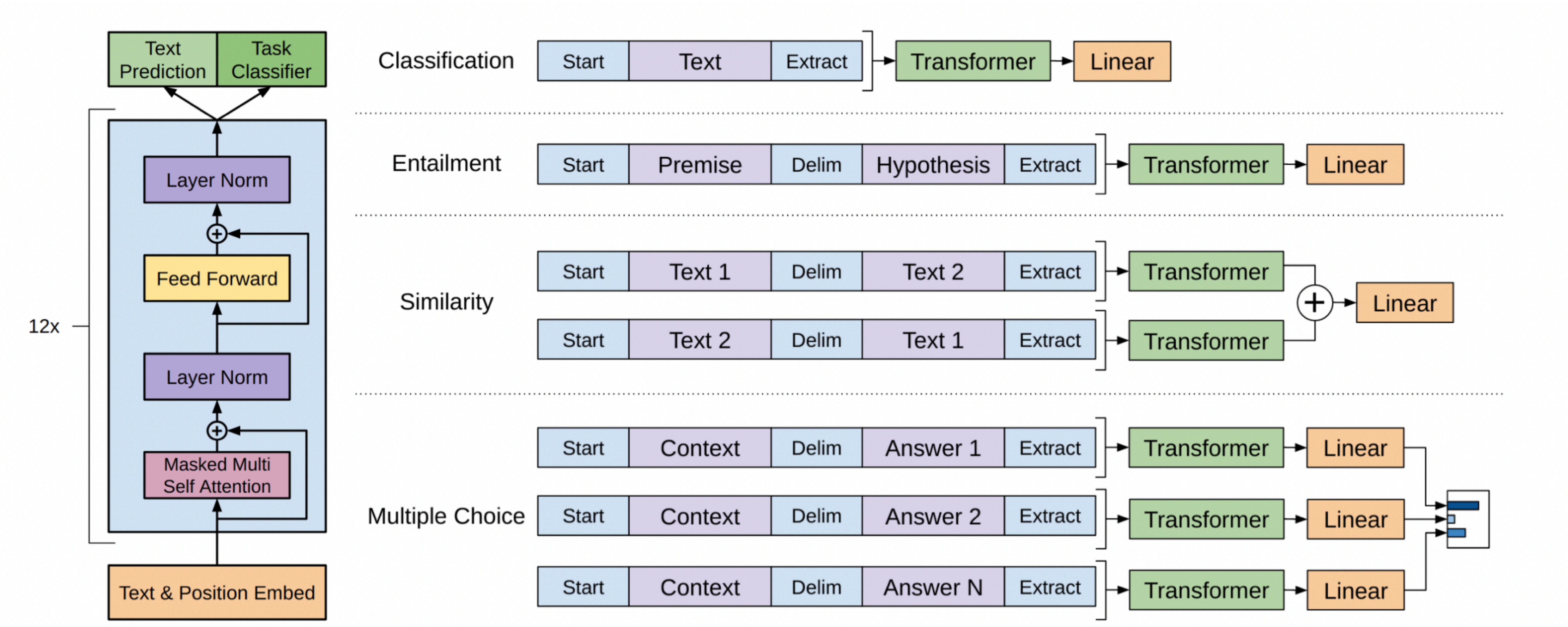
Formally, the language model objective models the joint distribution of a sequence of tokens: $(t_1,\ t_2,\ \ldots,\ t_n)$

$$p(t_1,\ t_2,\ \ldots,\ t_n) = \sum p(t_i \mid t_0, \ldots,\ t_{i-1})$$

It is a generative model: After training the language model, you can generate new sequences beyond the training data.

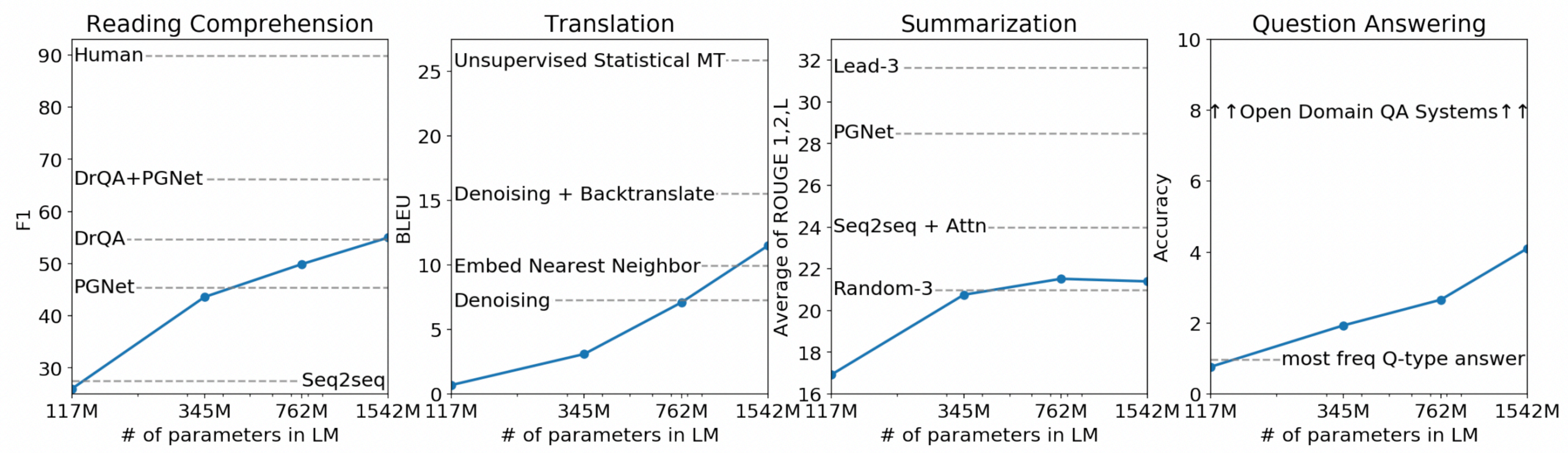
I. The Evolution of GPT Models of OpenAI

- GPT models of OpenAI
  - GPT-1: Improving Language Understanding by Generative Pre-Training. OpenAI. 2018.
  - GPT-2: Language Models are Unsupervised Multitask Learners. OpenAI. 2018.
  - GPT-3: Language Models are Few-Shot Learners. OpenAI. 2020.
  - GPT-4: GPT-4 Technical Report. 2023.
  - GPT-4o: Text, Audio, Image, and Video (“o” for “omni”). May, 2024.
  - OpenAI o1: Learning to Reason with LLMs. Sept, 2024.
- GPT-1 first uses the language model objective to pretrain Transformer, then finetunes the model on each downstream task.
- Data source: BooksCorpus dataset
  - Over 7,000 unique unpublished boo Romance.
- GPT-1 has 117 million parameters



I. The Evolution of GPT Models of OpenAI

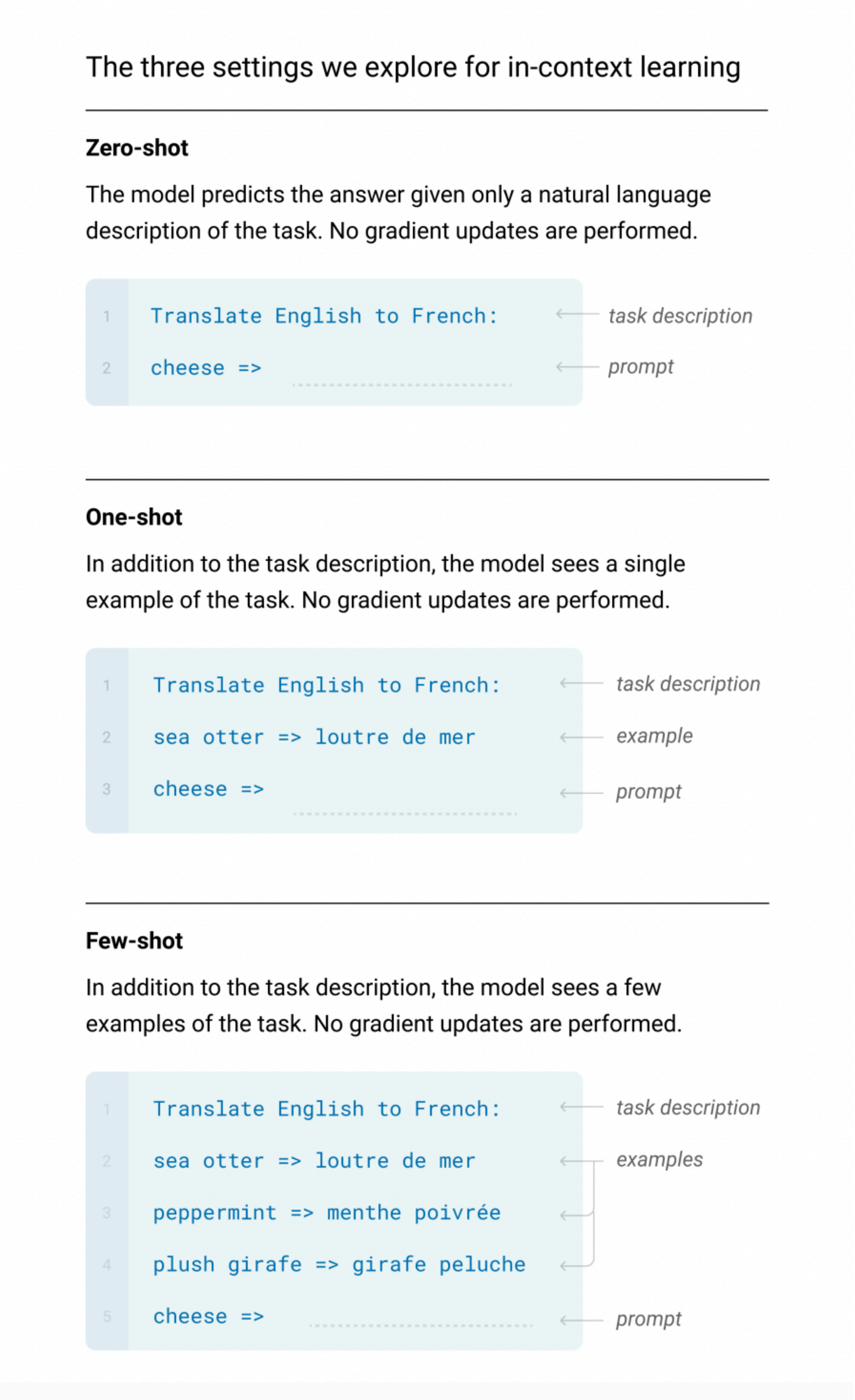
- GPT-2
  - Larger model size: The largest GPT-2 model has 1.5 billion parameters
  - More diverse data on the Web
  - Zero-shot task transfer
- Data source: WebText:
  - The outbound links from Reddit.
  - A total of 40 GB of text
- GPT-2 claims that as you pretrain a language model further with pretraining data, it will start to learn many NLP tasks automatically like question answering, machine translation, reading comprehension, and summarization.
- So they evaluate on these tasks in a zero-shot way after pretraining without finetuning.



Zero-shot performance on downstream tasks: As you enlarge the model, the performance on these tasks improves.

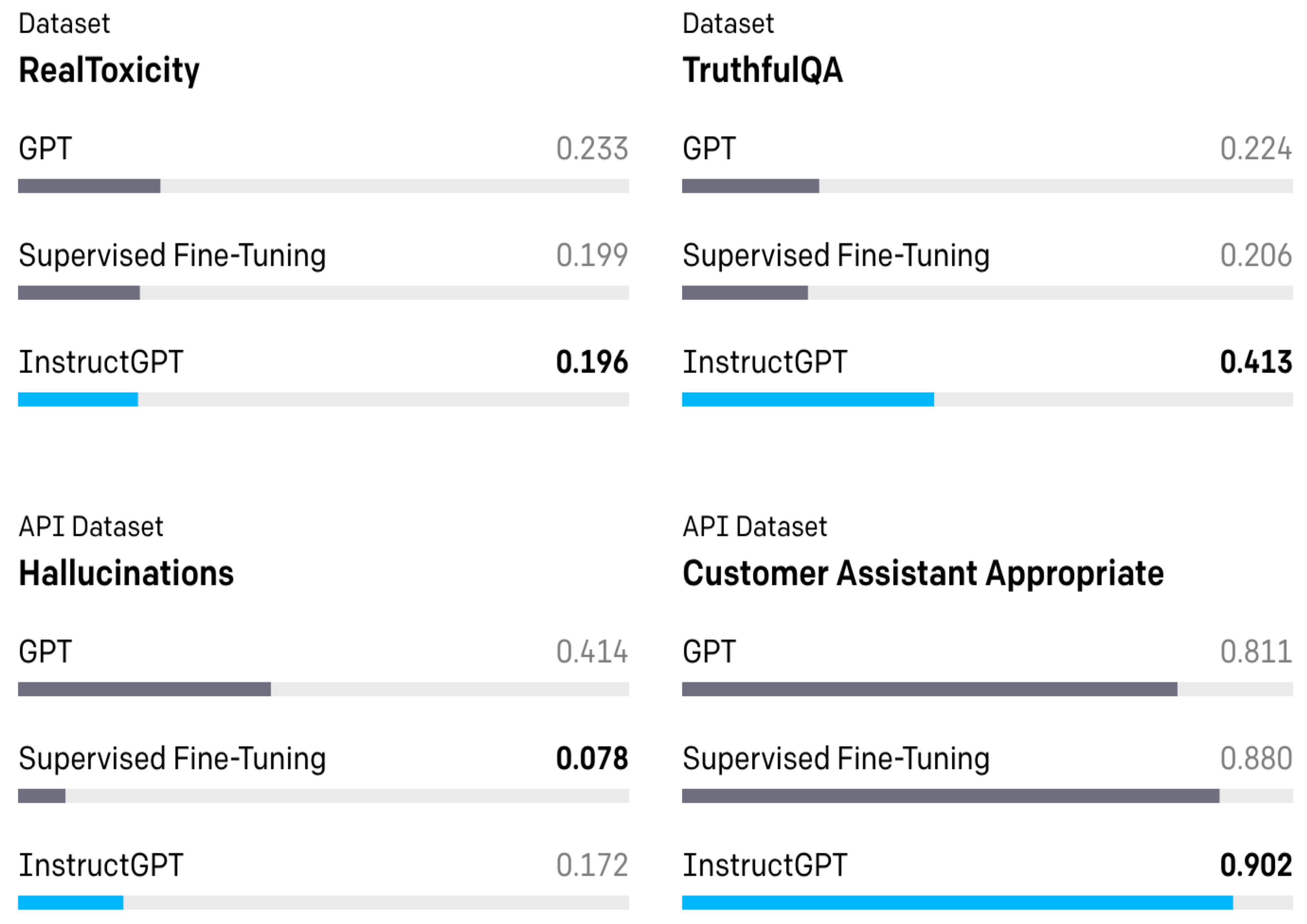
I. The Evolution of GPT Models of OpenAI

- **GPT-3**
  - Even larger model size: The largest GPT-3 model has 175 billion parameters
  - More data from the Web
  - Few-shot learning: a natural language instruction + a few examples without finetuning
- Data source:
  - CommonCrawl, WebText2, Books1, Books2, and Wikipedia
  - About 500 billion tokens; a total of 750+GB of text
- The text completion ability of GPT-3 is stronger than GPT-1 and GPT-2.
- Starting from GPT-3, OpenAI stops open-sourcing and just provides the APIs.
- Overall speaking, the GPT models are trained longer, use more diverse data, and scale the model size.



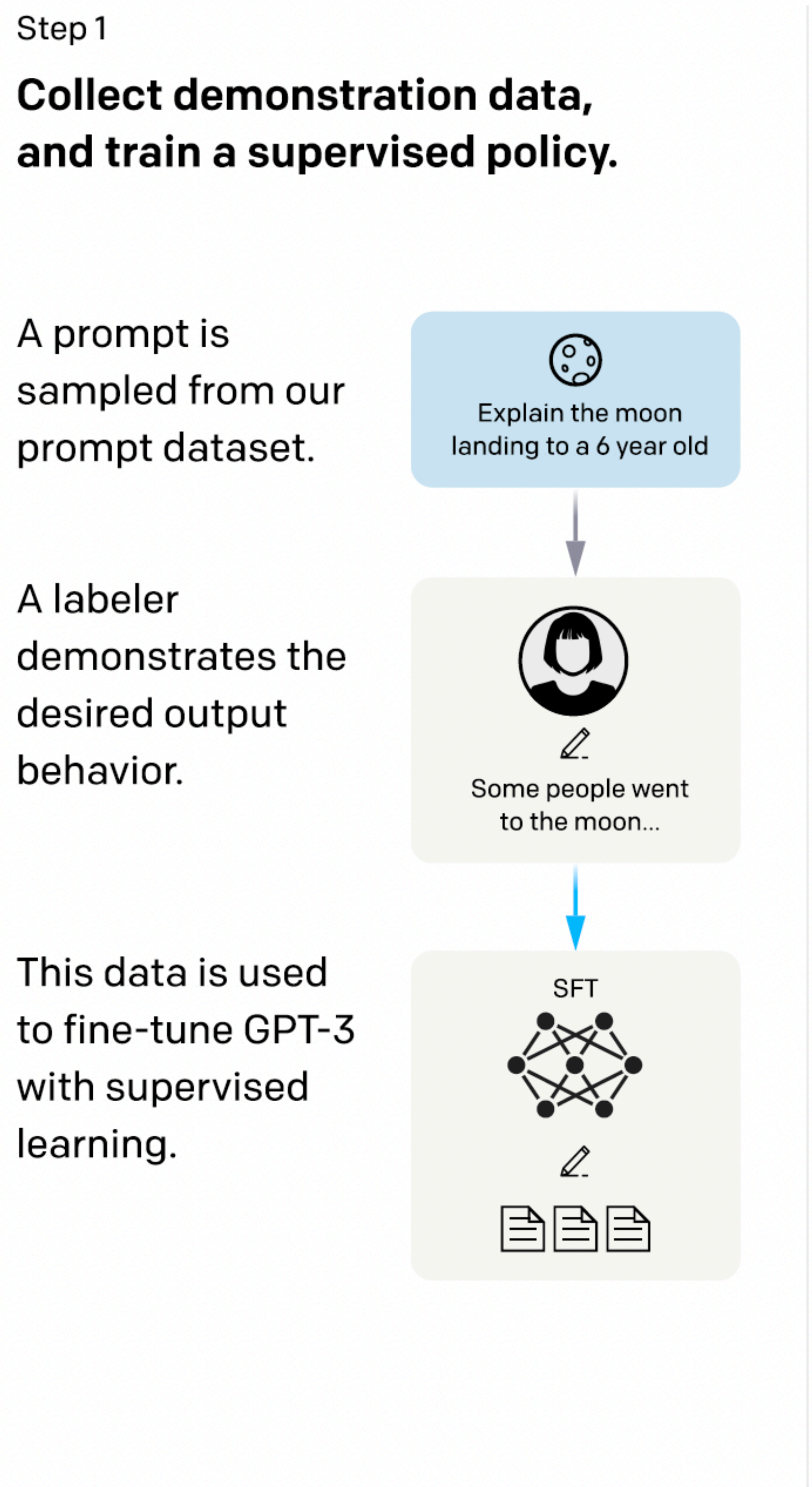
I. GPT-3.5/InstructGPT

- GPT-3 is a good text completion model.
- It served as APIs.
- It collects massive data from users about how users interact with GPT-3.
- OpenAI finds that the model trained with the language model objective is not good at following user intents.
- GPT-3.5 / InstructGPT:
  - Training language models to follow instructions with human feedback. OpenAI. 2022.
  - An improved version of the GPT-3.
  - Purpose: Train language models that are much better at following user intentions than GPT-3 while also making them more truthful and less toxic
- RLHF: Reinforcement Learning from Human Feedback
  - Incorporate human expertise and knowledge
  - Mitigate untruthful, toxic output and hallucination (making up information)
  - Make models more *aligned* with users



I. GPT-3.5/InstructGPT

- **GPT-3.5 / InstructGPT**: a fined-tuned version of the GPT-3
- **RLHF**: Reinforcement Learning from Human Feedback (Step 1: supervised fine-tuning (SFT))

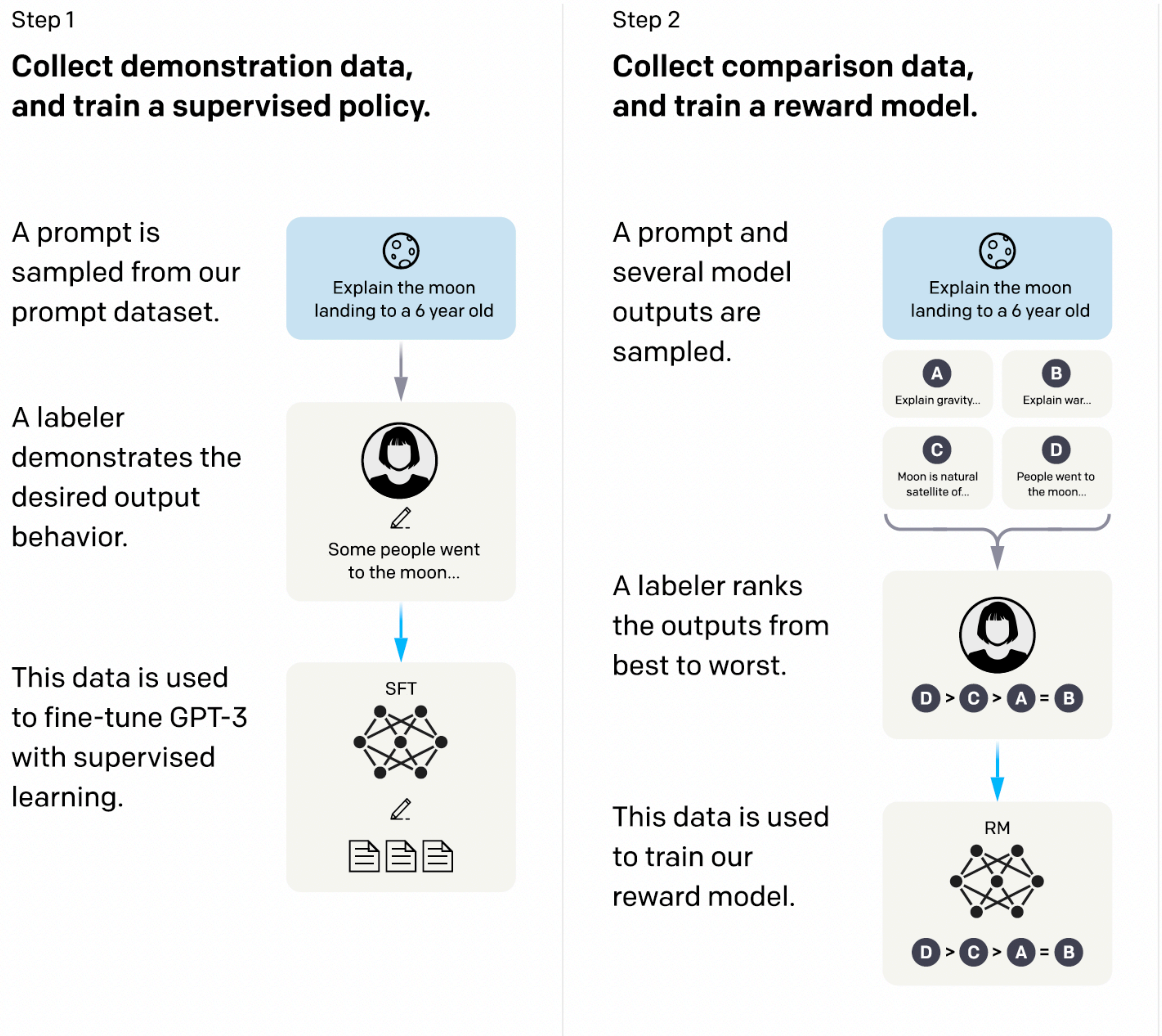


Requires human annotators to write desired output:

For example, given the input “what is the capital of the UK?”, the annotator needs to write the correct answer like “the capital of the UK is London”.

I. GPT-3.5/InstructGPT

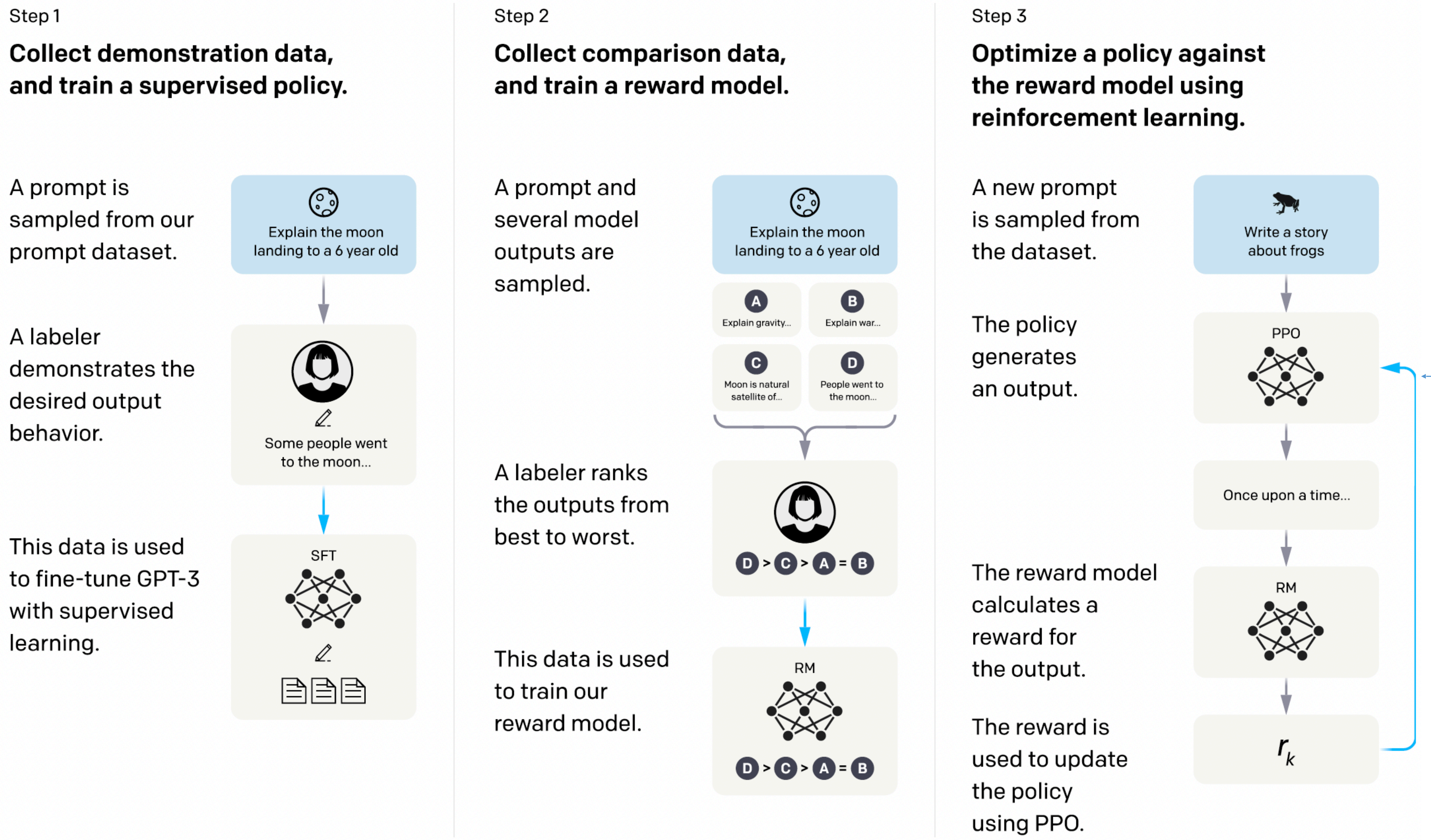
- **GPT-3.5 / InstructGPT**: a fined-tuned version of the GPT-3
- **RLHF**: Reinforcement Learning from Human Feedback (Step 2: reward model (RM) training)



Collecting human data is expensive and slow, the step 2 trains a reward model to replace human judgments.

I. GPT-3.5/InstructGPT

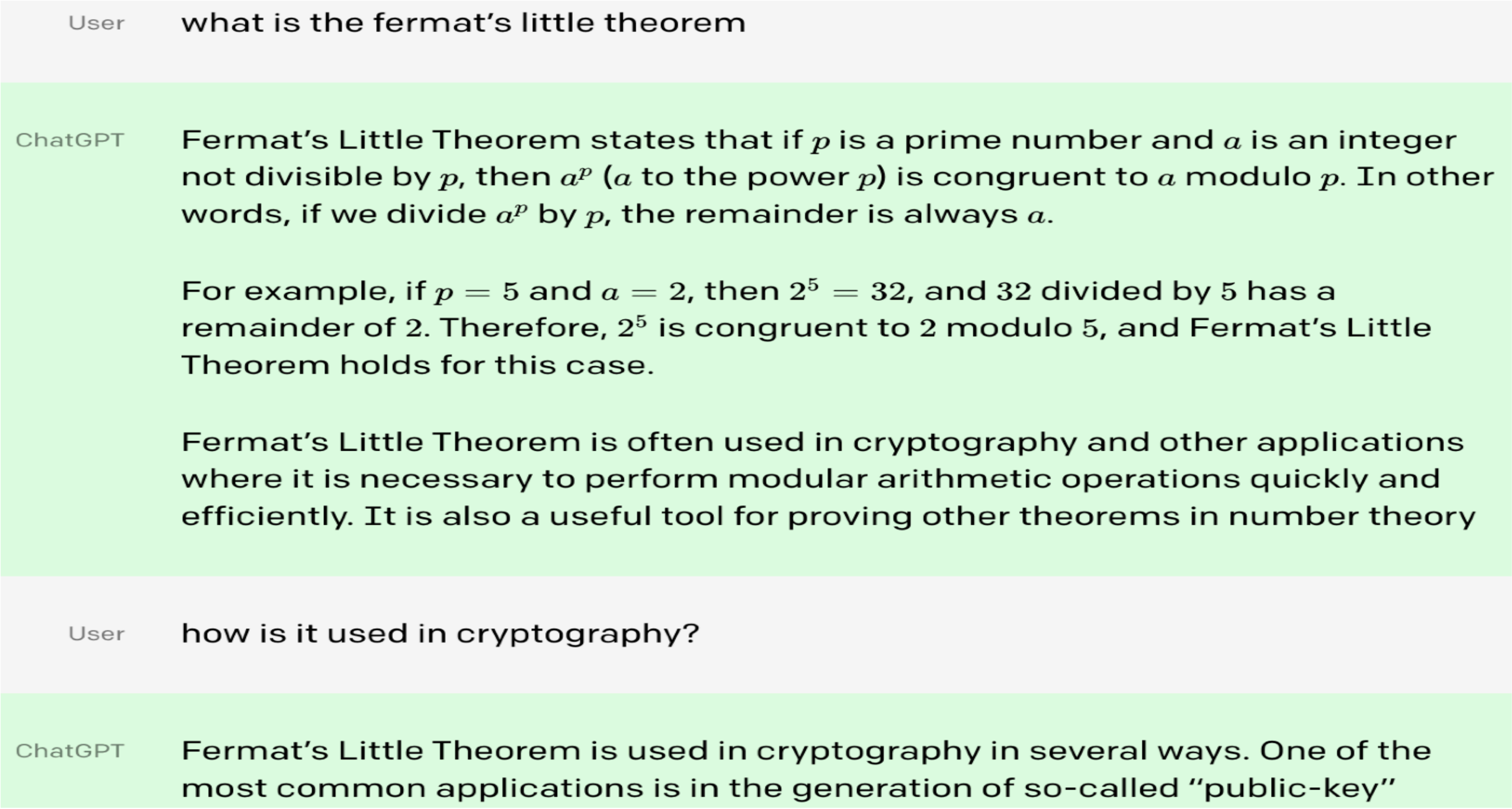
- **GPT-3.5 / InstructGPT**: a fined-tuned version of the GPT-3
- **RLHF**: (Step 3: reinforcement learning via proximal policy optimization (PPO))



Step 3 uses the model to interact with the reward model (a simulated human annotator). The reward model gives feedback to the model. Finally, the model is updated based on the feedback.

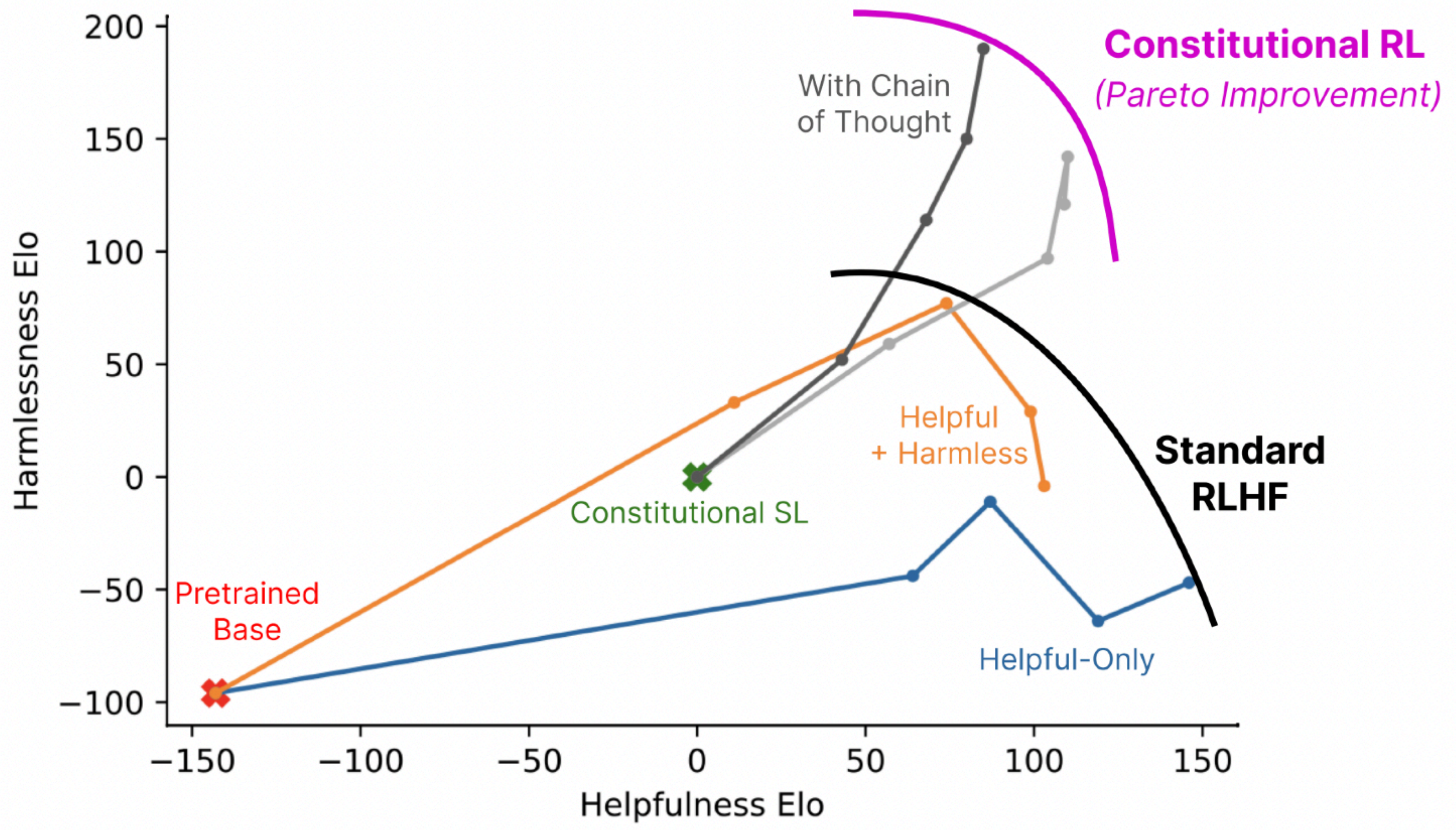
I. ChatGPT

- **ChatGPT**: based on InstructGPT, with slight differences in the data collection setup.
  https://chat.openai.com/ OpenAI. Nov 2022.
- Step 1: Train an InstructGPT model using Reinforcement Learning from Human Feedback (RLHF).
- Step 2: Human trainers use model-written suggestions to help them compose dialogue responses.
- Step 3: Mix this new dialogue dataset with the InstructGPT dataset in a dialogue format.



I. Helpfulness vs. Harmlessness

- For training ChatGPT, there is a tradeoff between helpfulness and harmlessness.
- During RLHF, if the reward stresses more on helpfulness, helpfulness tends to increase harmfulness.
- On the other hand, if the reward stresses more on harmlessness, the model tends to refuse to answer in many cases. This reduces helpfulness.
- Therefore, during RLHF, we need to achieve a balance between helpfulness and harmlessness.



Standard RLHF is helpful but also harmful. Need to strike a balance between helpfulness and harmlessness.

I. The Importance of Training Data

- The idea of instruction tuning used by Instruct-GPT is not new.
- FLAN already use this technique before Instruct-GPT.
- We note that the prompts of Instruct-GPT are collected from GPT-3 APIs, yet FLAN use public NLP datasets.
- The performance of FLAN underperforms Instruct-GPT.
- This highlights
  - (1) The importance of training data.
  - (2) The importance of building the flywheel effect of data.
- If we suppose OpenAI open sourced GPT-3 in 2020,
  - It cannot collect massive user data.
  - The competitive advantage of OpenAI may disappear.

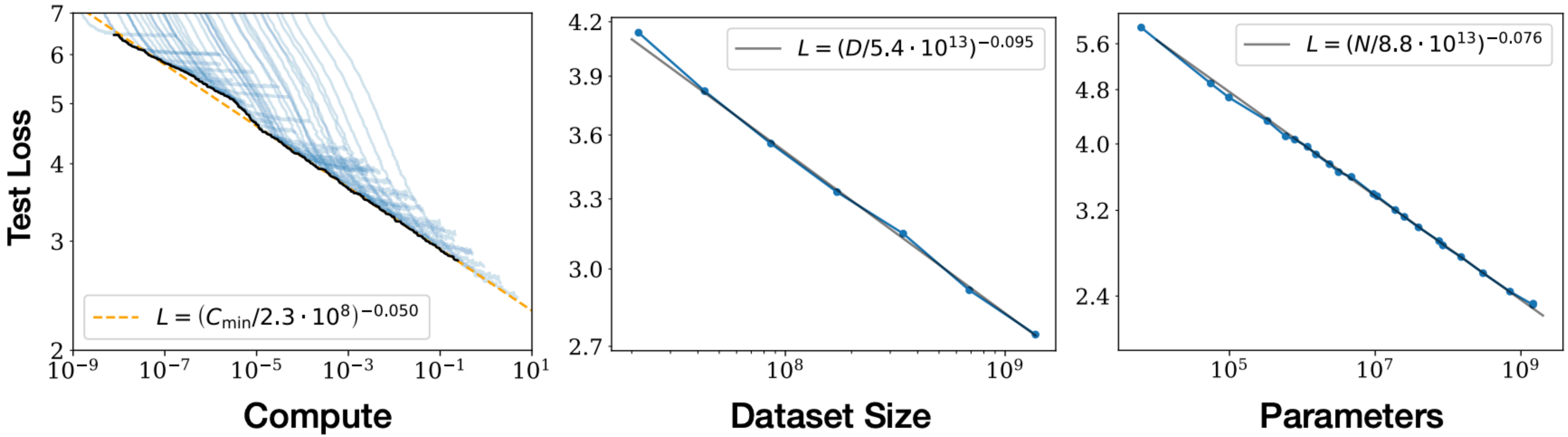
In retrospect, since many other models just fine-tune on GPT generated data, this may starts a lot of things.

II. Three Key Components of Training LLMs

- The three key components of training LLMs include
  - Training Data
  - Model scale
  - Compute budget
- We can simply scale the size of Transformer by increasing the number of self-attention layers and the hidden size.
  - Increasing the number of parameters.
- However, training data and compute budget are equally important for achieving good performance.
  - A fact is that most large language models are significantly undertrained: they are not trained long enough.
  - Curating a pretraining dataset takes time.
  - Challenge: We are exhausting the Web data to pretrain large language models.

II. Compute, Dataset Size, and Model Scale

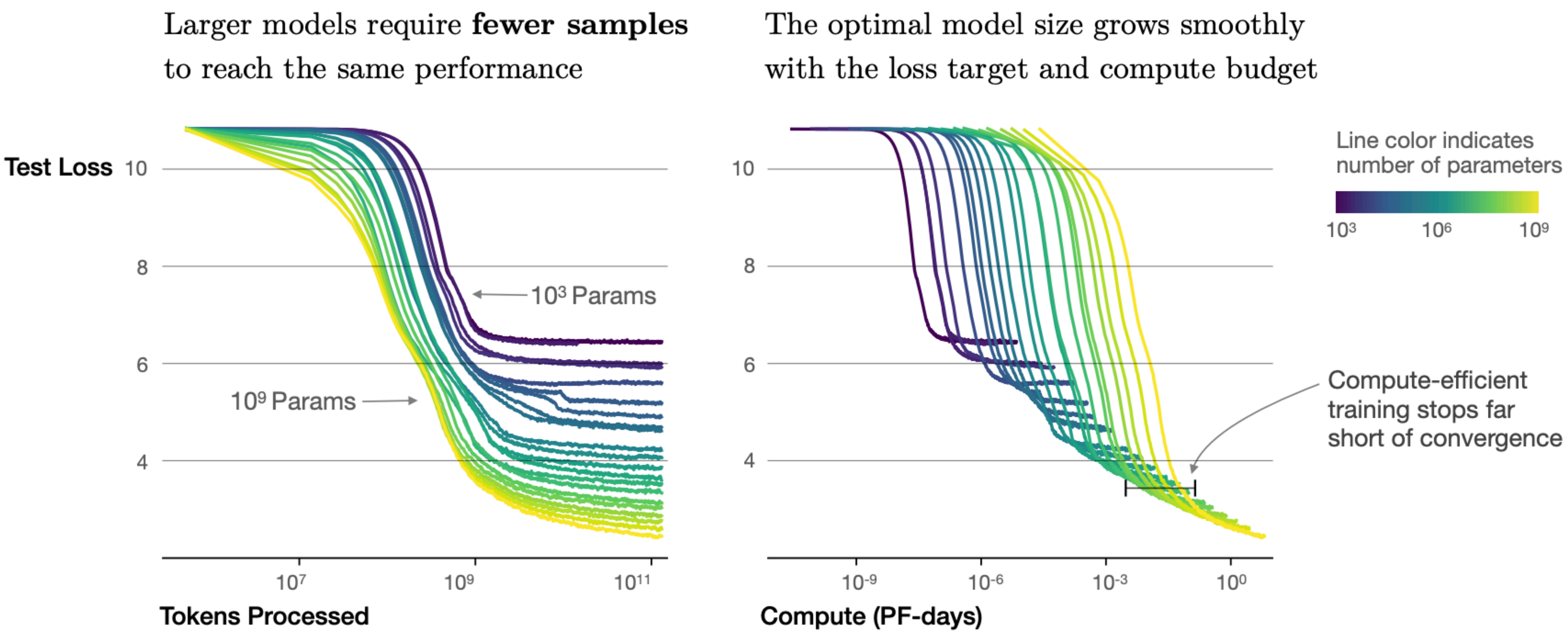
- Language modeling performance **improves smoothly** as we increase the model size, dataset size, and the amount of compute used for training.



Scaling Laws for Neural Language Models. OpenAI. 2020.

II. Compute, Dataset Size, and Model Scale

- Large models are more **sample-efficient** than small models, reaching the same level of performance with **fewer optimization steps**.



Scaling Laws for Neural Language Models. OpenAI. 2020.

II. Compute-Optimal Large Language Models

- How many data do we need to train a large model?

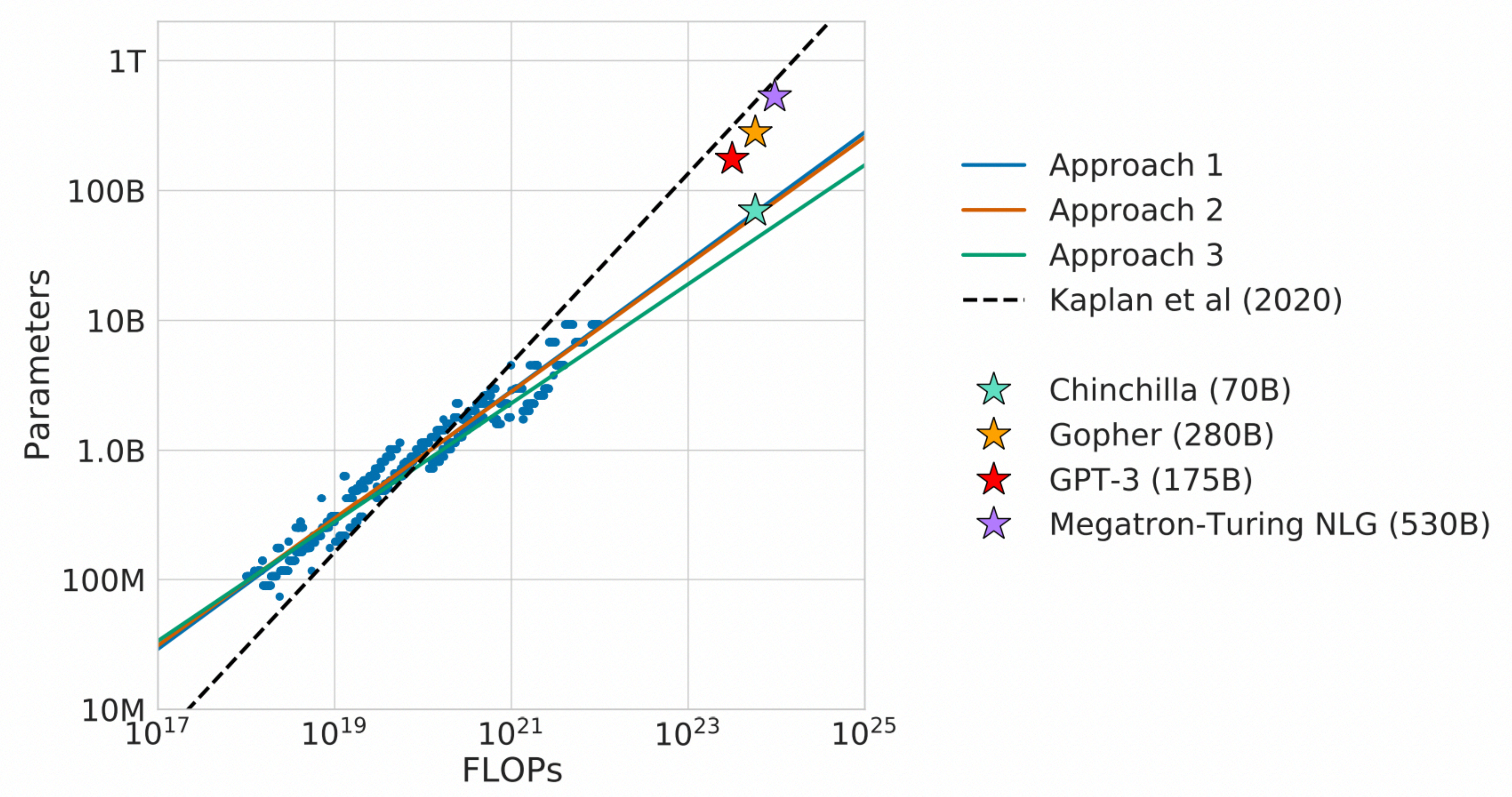
- The model size and the number of training tokens should **be scaled equally**: for every doubling of model size the number of training tokens should also be doubled.
  - Therefore, training a model that is too large is not necessary if there is not enough training data.

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

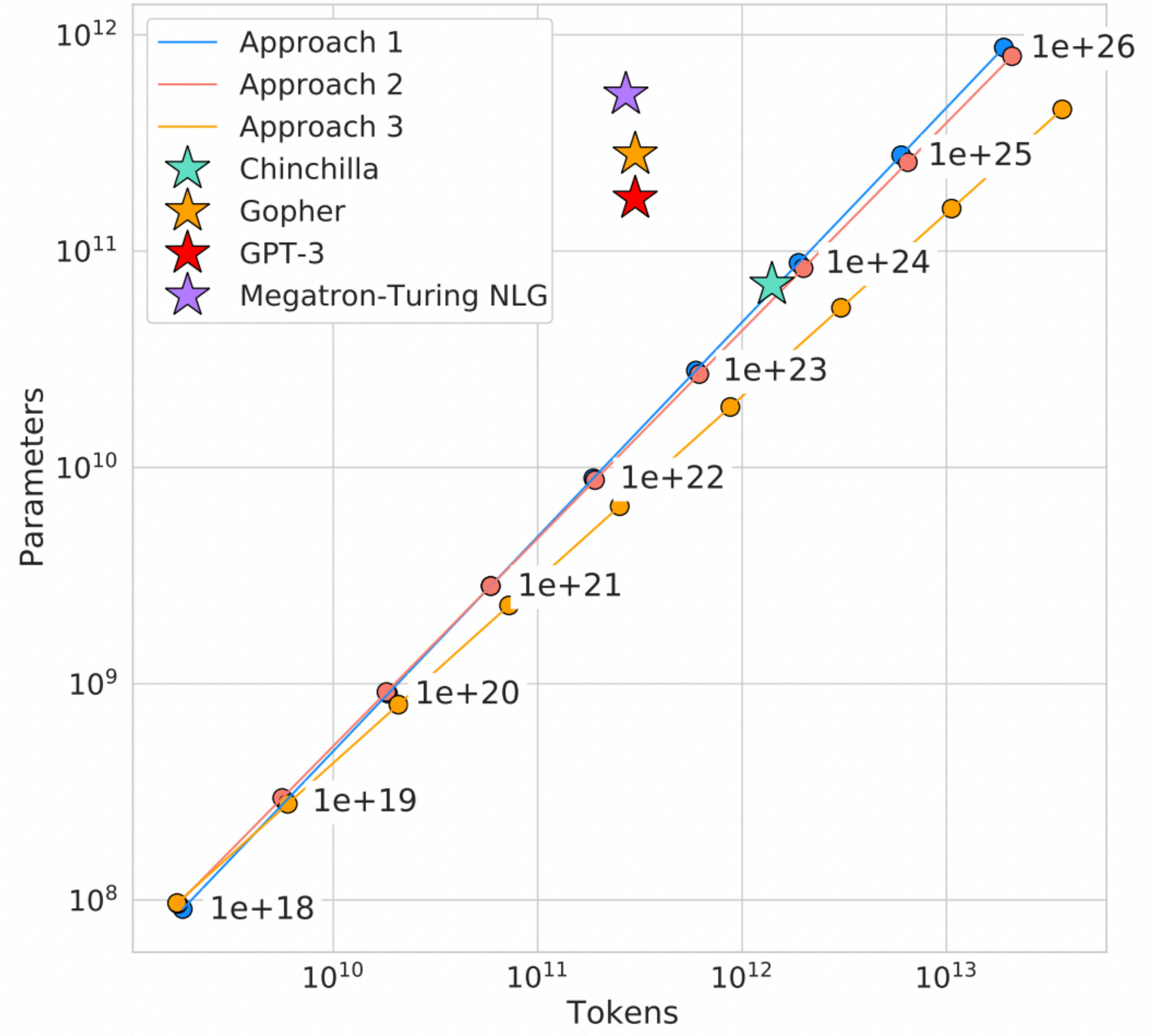
Training Compute-Optimal Large Language Models. DeepMind. 2022.

II. Compute-Optimal Large Language Models

- Current large models should be substantially smaller and therefore trained much longer than is currently done.



Optimal Compute Given the Model Scale

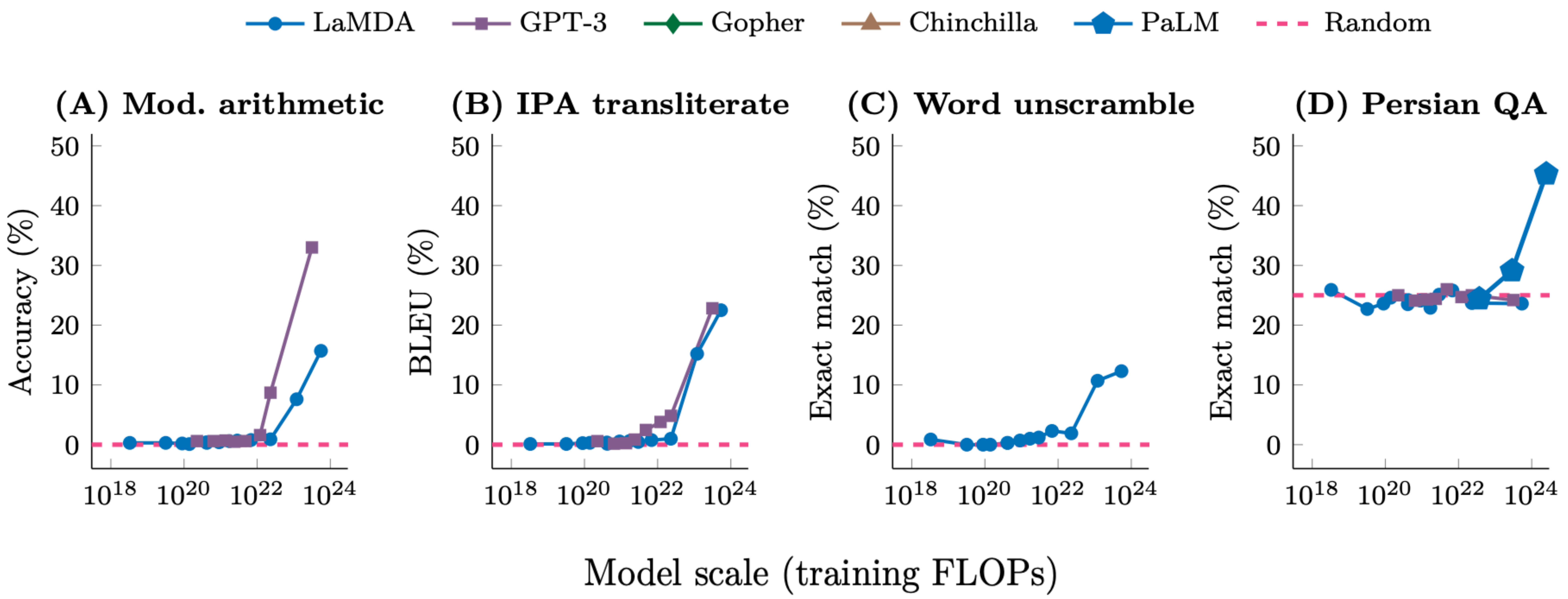


Optimal Data Size Given the Model Scale

Training Compute-Optimal Large Language Models. DeepMind. 2022.

III. Emergent Abilities

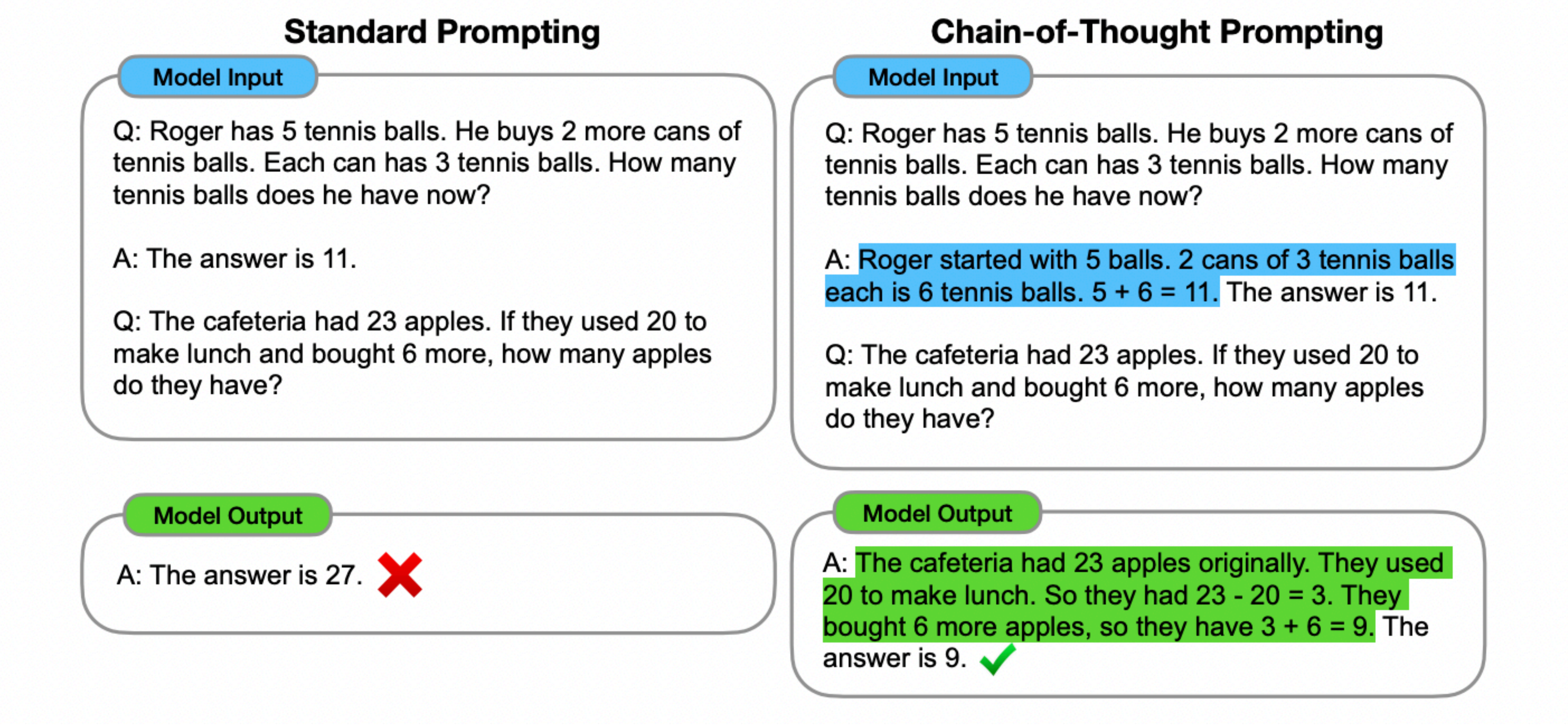
- **Emergent Abilities** of large language models: an ability to be emergent if it is not present in smaller models but is present in larger models.



Emergent Abilities of Large Language Models. Google. 2022.

III. Chain of Thought

- Language language model can generate a chain of thought—a series of intermediate reasoning steps.
- It can significantly improve the ability of large language models to perform complex reasoning.
- Chain of though abilities exhibit naturally in large language models.



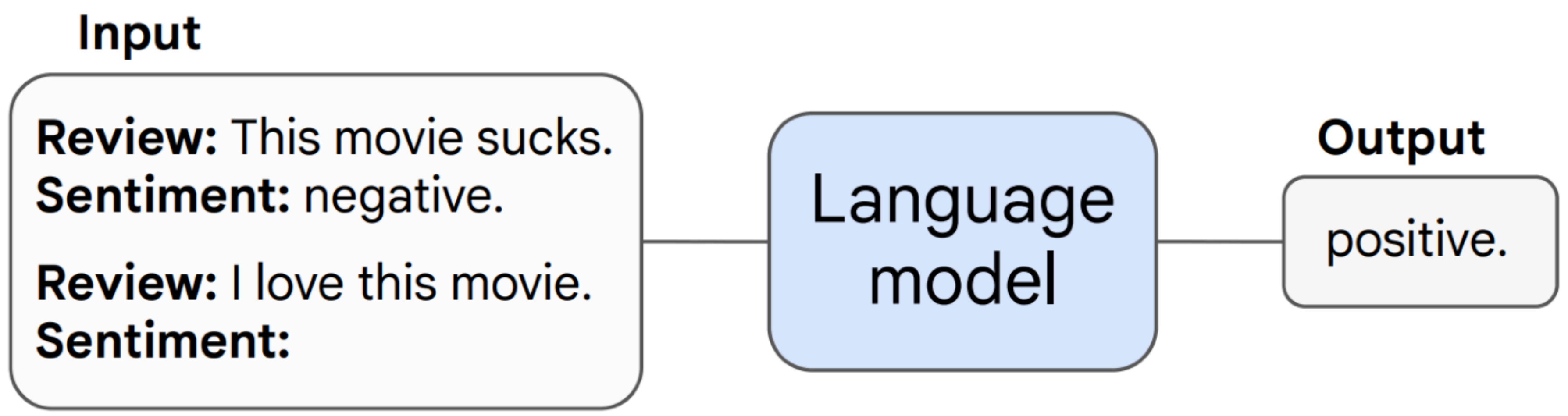
Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Google. 2022.

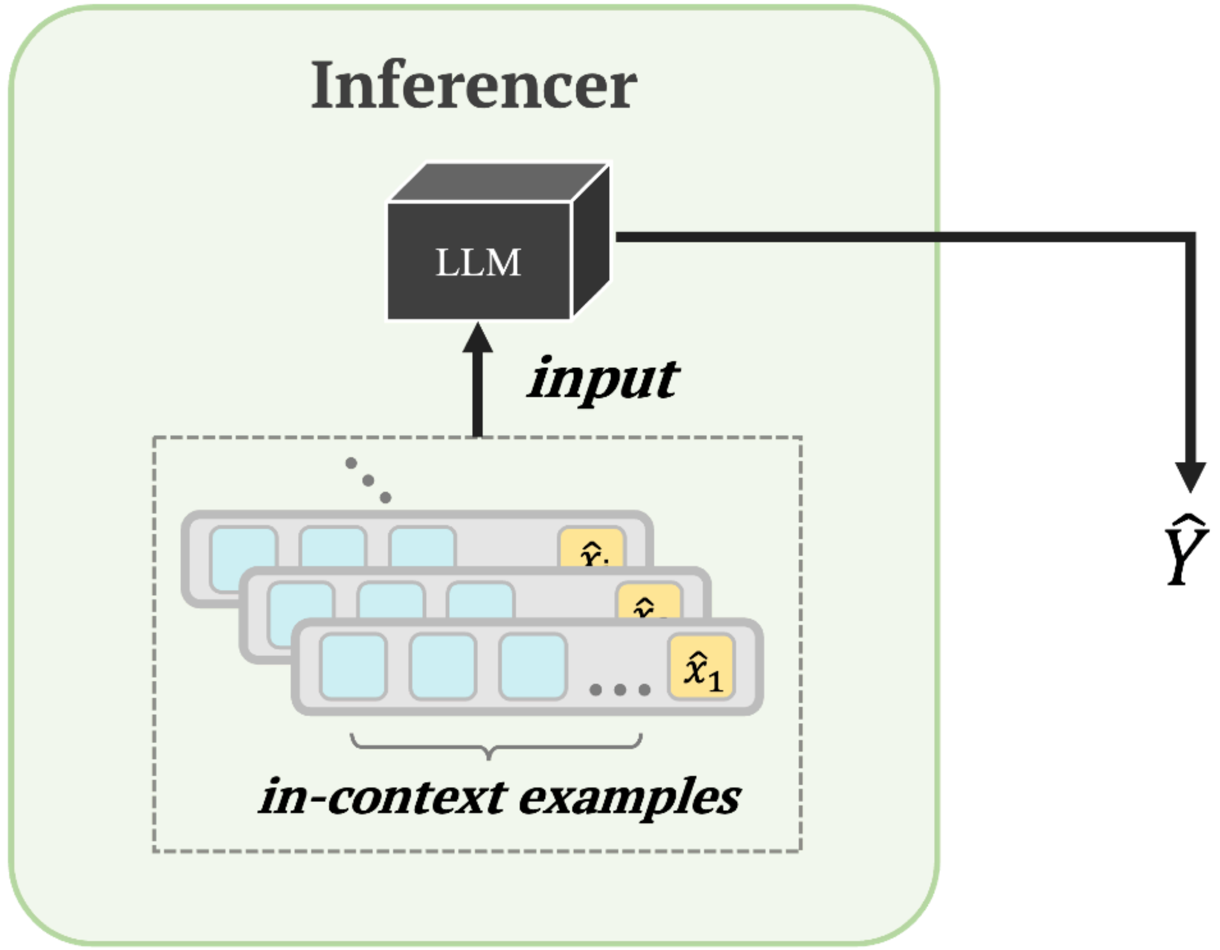
III. In-Context Learning

- Autonomous Action: LLMs as agents can perform tasks independently by interpreting instructions and taking actions based on the given context, simulating decision-making processes.

- Interactive Capabilities: They engage in dynamic interactions, adapting their responses and actions based on real-time input, enabling more complex task execution.

- Multi-Tasking: These agents can handle a variety of tasks across domains, from answering queries to navigating workflows, by leveraging their understanding of language and context.



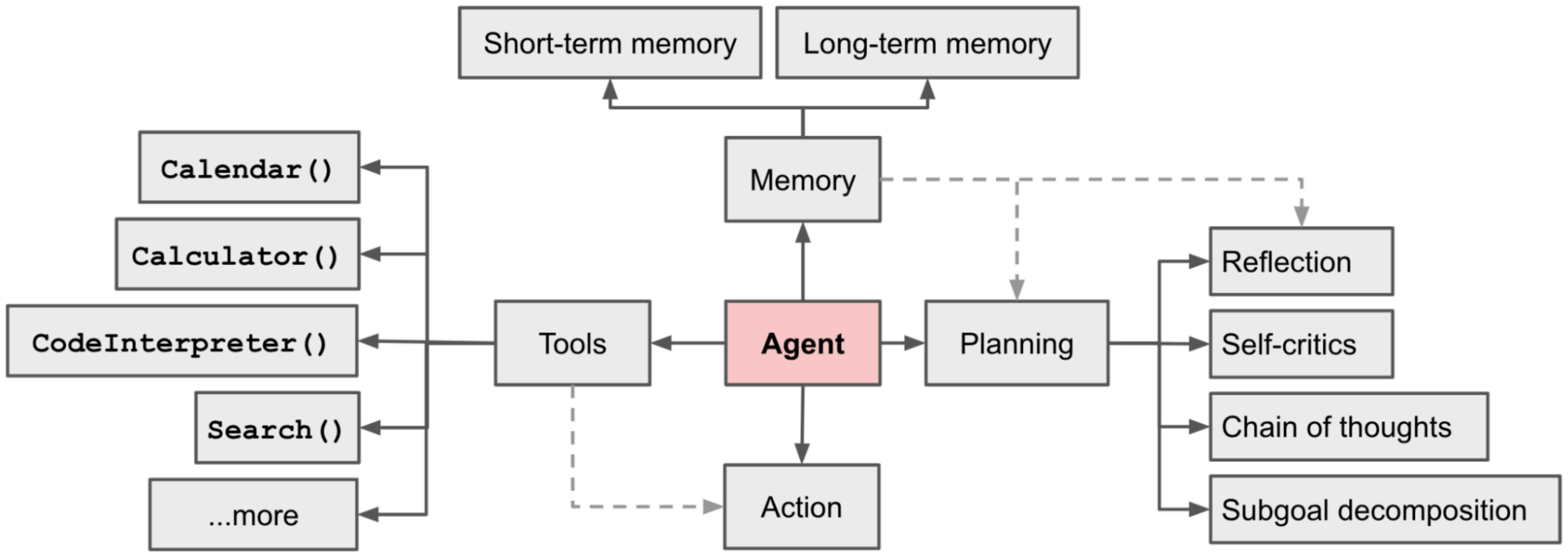


In-Context Learning

A Survey on In-context Learning. Qingxiu Dong, et al. 2022

III. LLMs as Agents

- Autonomous Action: LLMs as agents can perform tasks independently by interpreting instructions and taking actions based on the given context, simulating decision-making processes.
- Interactive Capabilities: They engage in dynamic interactions, adapting their responses and actions based on real-time input, enabling more complex task execution.
- Multi-Tasking: These agents can handle a variety of tasks across domains, from answering queries to navigating workflows, by leveraging their understanding of language and context.



https://lilianweng.github.io/posts/2023-06-23-agent/

III. LLMs as Agents

HotpotQA

(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]

AlfWorld Env

(2b) ReAct (Reason + Act)

Act 1: **Think**[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ...]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.

(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)

Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1, ... a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

Act 8: **Think**[Now I find a pepper shaker 1. Next, I need to put it in/on drawer 1.]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 …

Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.

https://lilianweng.github.io/posts/2023-06-23-agent/