HSLU Lucerne University of Applied Sciences and Arts

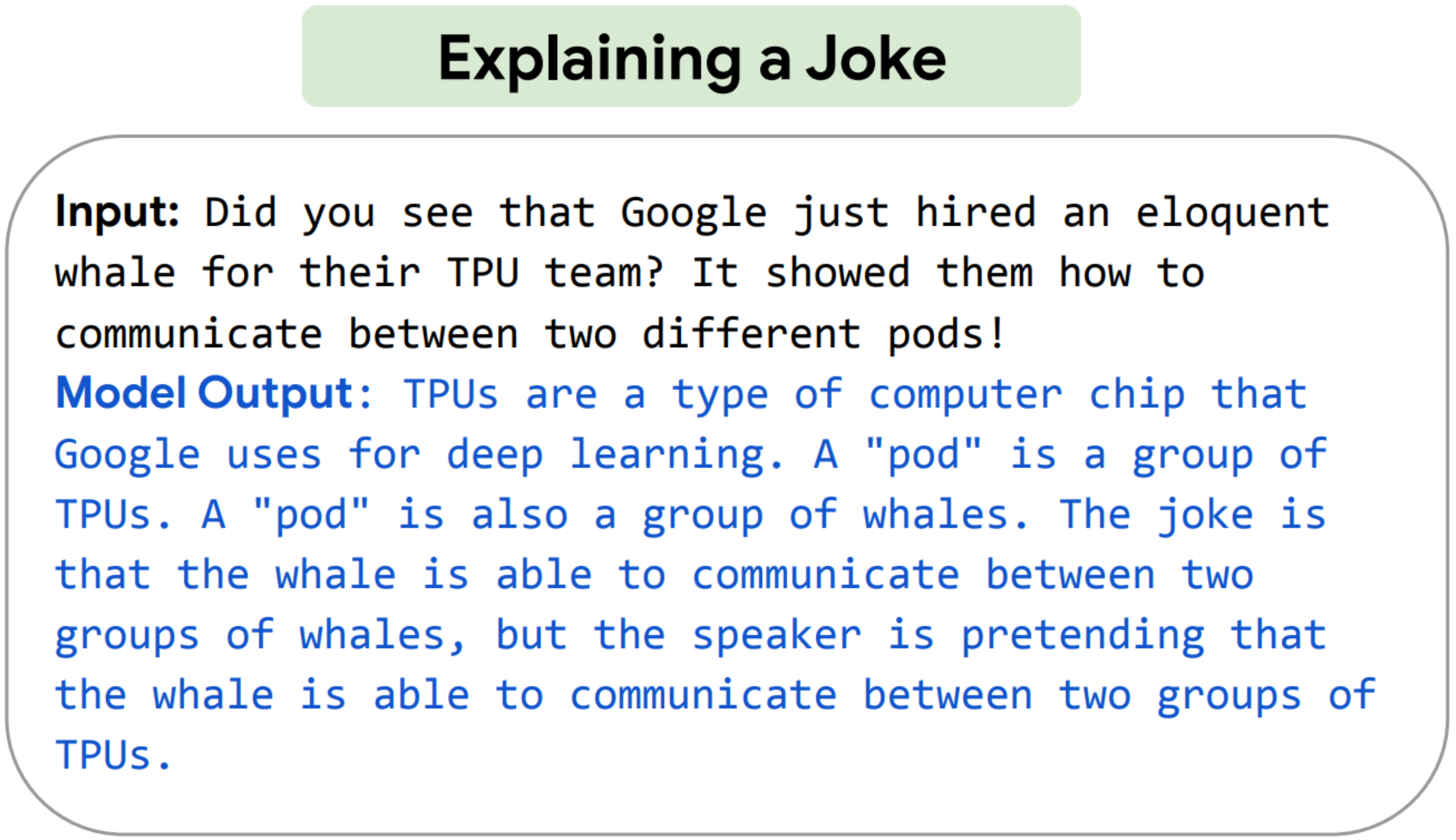
Pretrained Language Models

NLP
Andreas Marfurt

Information Technology
24.04.2025

FH Zentralschweiz

Motivation

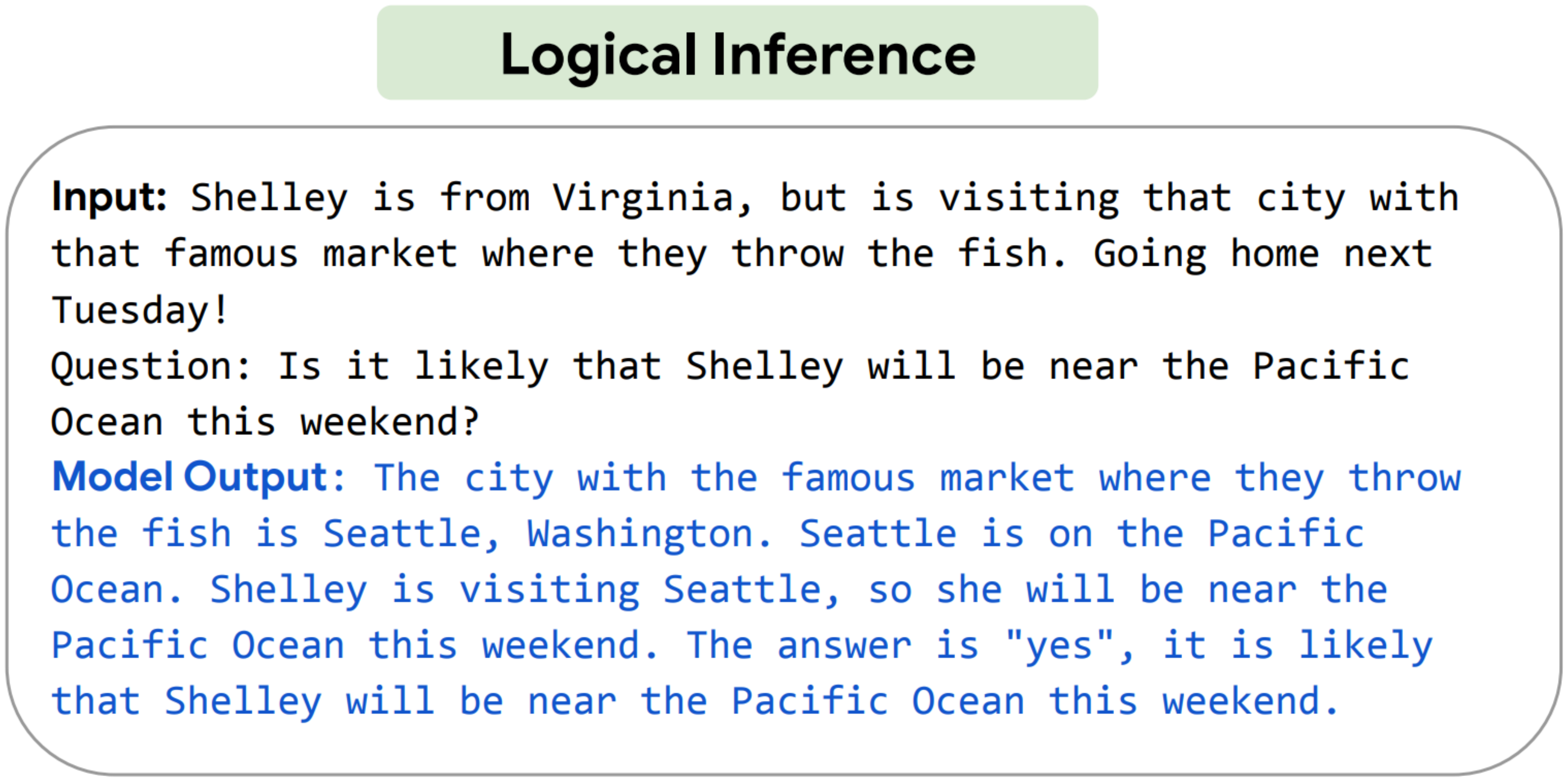


Explaining a Joke

Input: Did you see that Google just hired an eloquent whale for their TPU team? It showed them how to communicate between two different pods!

Model Output: TPUs are a type of computer chip that Google uses for deep learning. A "pod" is a group of TPUs. A "pod" is also a group of whales. The joke is that the whale is able to communicate between two groups of whales, but the speaker is pretending that the whale is able to communicate between two groups of TPUs.

Motivation



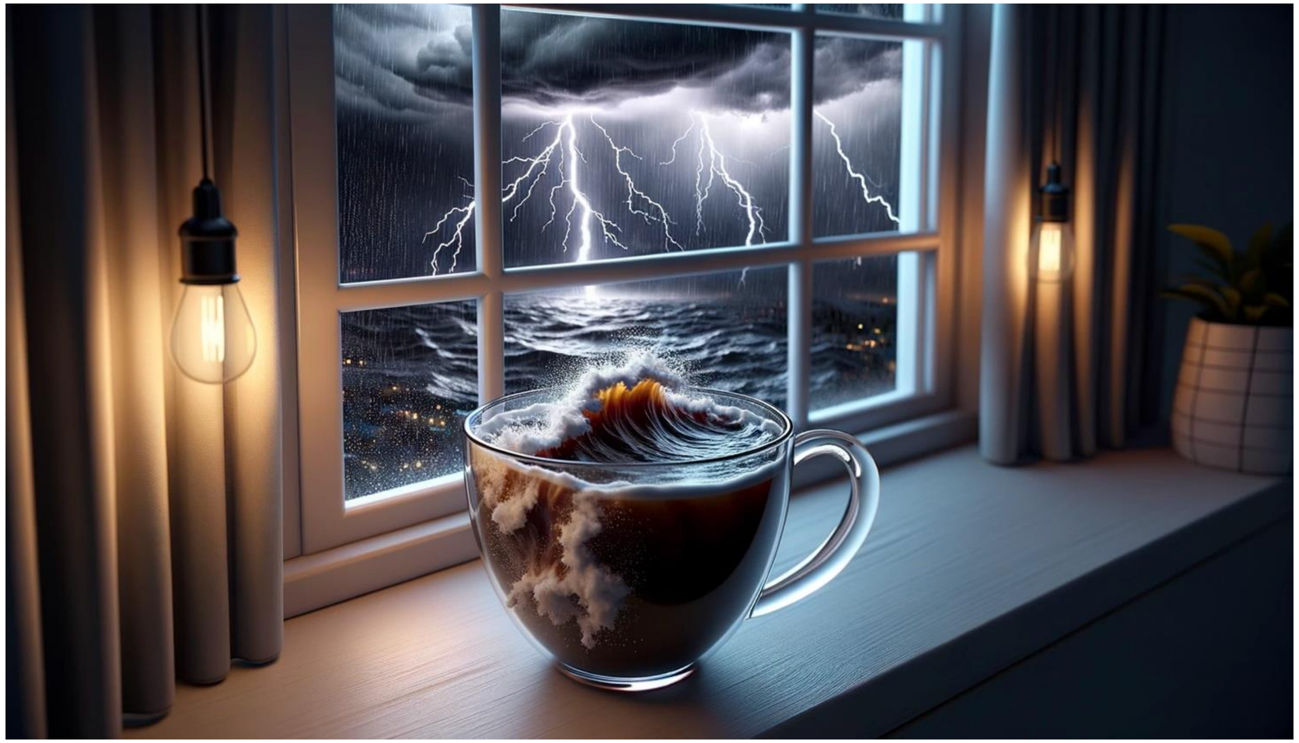
Logical Inference

Input: Shelley is from Virginia, but is visiting that city with that famous market where they throw the fish. Going home next Tuesday!
Question: Is it likely that Shelley will be near the Pacific Ocean this weekend?
Model Output: The city with the famous market where they throw the fish is Seattle, Washington. Seattle is on the Pacific Ocean. Shelley is visiting Seattle, so she will be near the Pacific Ocean this weekend. The answer is "yes", it is likely that Shelley will be near the Pacific Ocean this weekend.

Motivation

“Teddy bears mixing sparkling chemicals as mad scientists as a 1990s Saturday morning cartoon”

Motivation



“A 3D render of a coffee mug placed on a window sill during a stormy day. The storm outside the window is reflected in the coffee, with miniature lightning bolts and turbulent waves seen inside the mug. The room is dimly lit, adding to the dramatic atmosphere.”

Overview

- Pretrained Encoder-Decoder Models
  - BART
  - T5
- Pretrained Decoder-only Models
  - GPT
  - Prompting
    - Classifying with Language Models
    - Zero- vs. Few-shot Prompting
    - Prompt Engineering and Chain-of-thought Prompting
  - Instruction Tuning
  - Alignment

Why are encoders not all we need?

- Autoregressive (“left-to-right”) generation is the dominant text/sequence generation method
- Non-autoregressive decoding exists (e.g. Insertion Transformer)
  - Easy to parallelize → faster
  - But does not work well (…yet?)

Text Generation

Text Generation is useful for a lot of tasks:

- Machine translation
- Question answering
- Summarization
- Text simplification
- Dialogue/chatbots
- … and other tasks can be reformulated as a text generation task (as we will see soon)

Bonus *Explainability*: We can ask the model to explain its decisions with language

Encoder-Decoder Models

- We know those already:
  - Seq2seq RNN
  - Transformer
- Encoder
  - *Encodes* an input → Generates a (usually simplified) representation of the input
  - Bidirectional attention
- Decoder
  - Generates an output
    - Class
    - Score
    - Text
  - Causal attention (can't see the future)

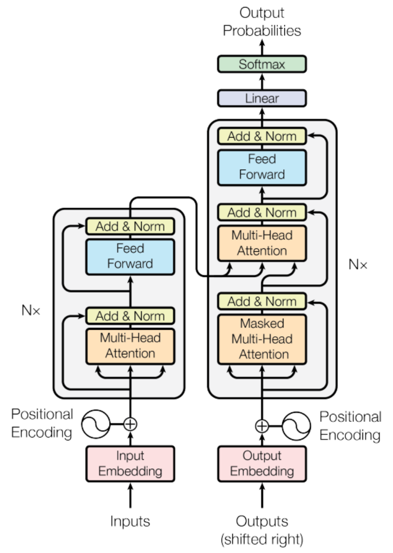
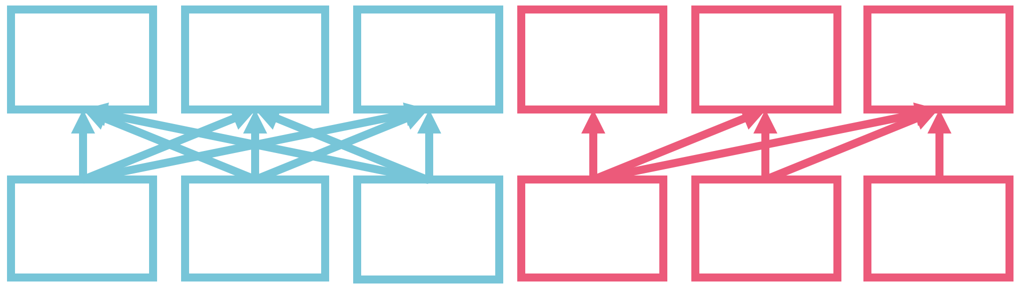
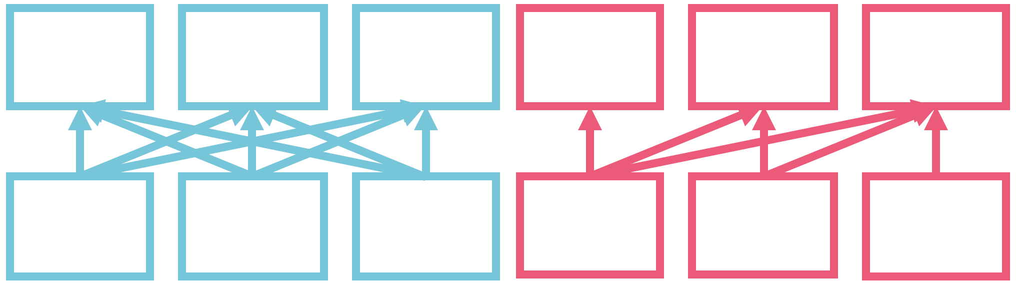


Figure 1: The Transformer - model architecture.



Pretraining Encoder-Decoder Models

- What pretraining objective should we use?
  - Train the encoder for document understanding
  - Train the decoder for text generation
  - Train the encoder representation to be useful for the decoder
    → Help the decoder generate text that is relevant to the input



- Do we get all of this from MLM?

BART

- Lewis et al., 2020
- BART = Bidirectional and Auto-Regressive Transformer
- Denoising autoencoder

Add noise to the input, then reconstruct the original text.

BART Pretraining Objectives

Original text: I saw a cat. It was dark.

Noise transformations

- Token masking: I ___ a cat. ___ was dark.
- Token deletion: I a cat. dark.
- Text infilling: I ___ . It was ___ dark.
- Sentence permutation: It was dark. I saw a cat.
- Document rotation: was dark. I saw a cat. It

= MLM
(same as BERT)

BART Findings

- Best pretraining method depends on the task
  - This is not what we wanted to hear: We would like to pretrain once and then just finetune on the task
  - It turns out that a good pretraining does a reasonably good job at transferring the knowledge
- Token masking is necessary
  - Just permutation/rotation leads to bad performance
- MLM and permutation perform worse on generation
- Bidirectional encoders are important for question answering
- The pretraining objective is not the only important factor
  - Architecture decisions matter as well

BART Model

- Use text infilling and sentence permutation
  - Mask 30% of tokens
  - Permute all sentences
- 12 layers in encoder and decoder each
- 1024 hidden_dim

Using the BART Model

```
>>> from transformers import BartTokenizer, BartModel

>>> tokenizer = BartTokenizer.from_pretrained('facebook/bart-large')
>>> model = BartModel.from_pretrained('facebook/bart-large’)

>>> inputs = tokenizer("Hello, my dog is cute", return_tensors="pt")
>>> outputs = model(**inputs)
>>> last_hidden_states = outputs.last_hidden_state
```

Multilingual BART (mBART)

- Liu et al., 2020
- Pretrain model on denoising text in 25 languages
- Finetune for machine translation
- Use language codes as a prefix/suffix
- Pretraining format:
  Text in English [EOS] [EN] → [EN] Text in English [EOS]
- Translation format (during finetuning):
  Text in English [EOS] [EN] → [JA] Text in Japanese [EOS]

mBART-50

- Tang et al., 2020
- Extend mBART to 50 languages
- Use same format everywhere:

  [language code] text [EOS]

- Most popular mBART version: facebook/mbart-large-50

T5 (Text-to-text Transfer Transformer)

- Raffel et al., 2020
- Unified model for many tasks by formulating them as text-to-text tasks
- Large-scale study of different design decisions
  - Pretraining objectives
  - Architectures
  - Pretraining datasets
  - Transfer approaches
- Collected C4 (= "Colossal Clean Crawled Corpus")
- Evaluated on many language understanding tasks

T5: Tasks in Text Format

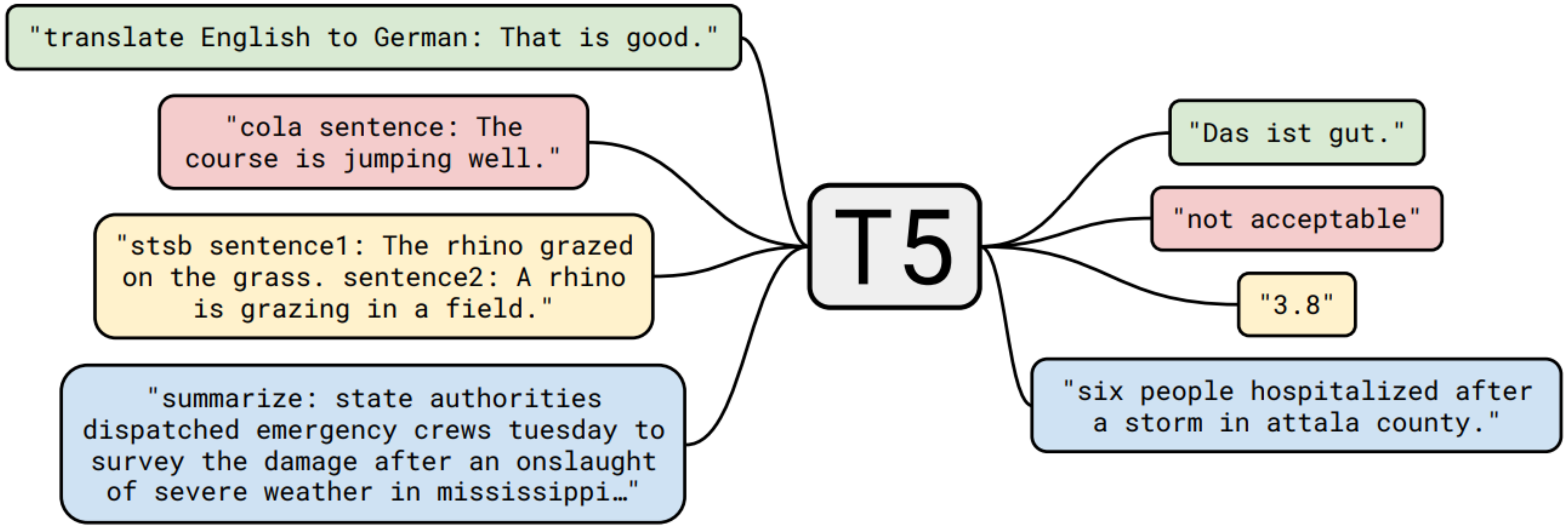
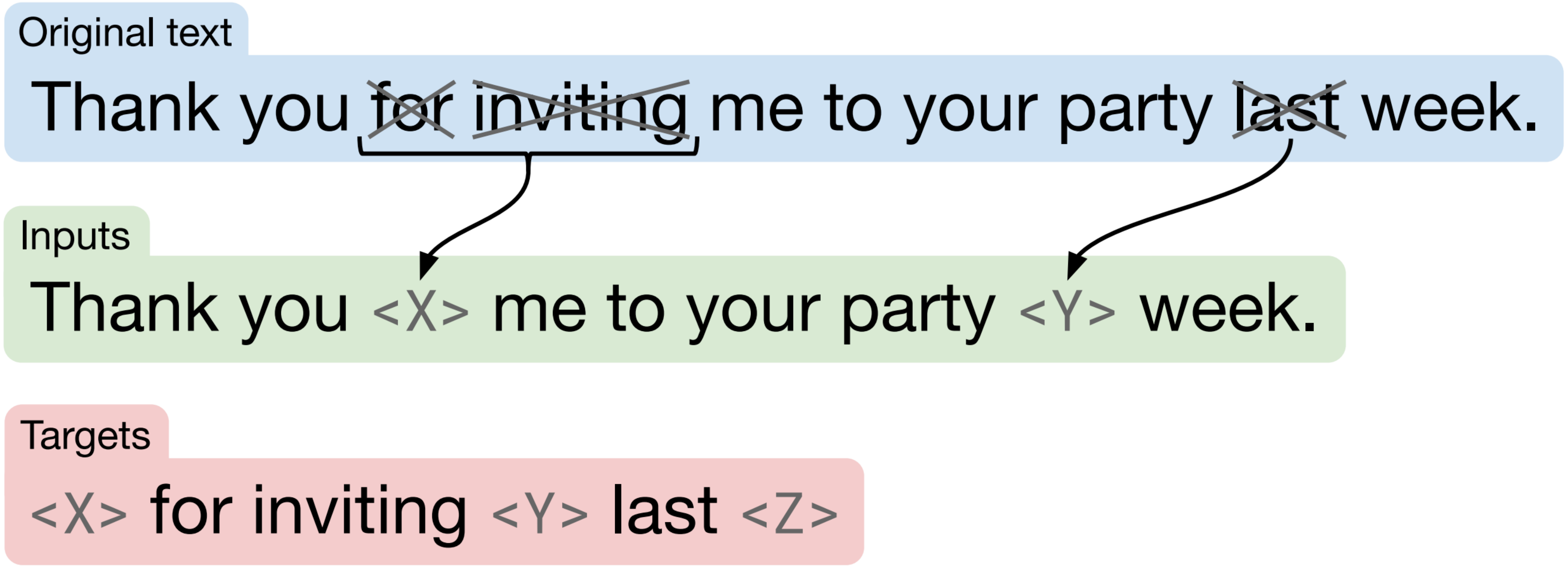

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

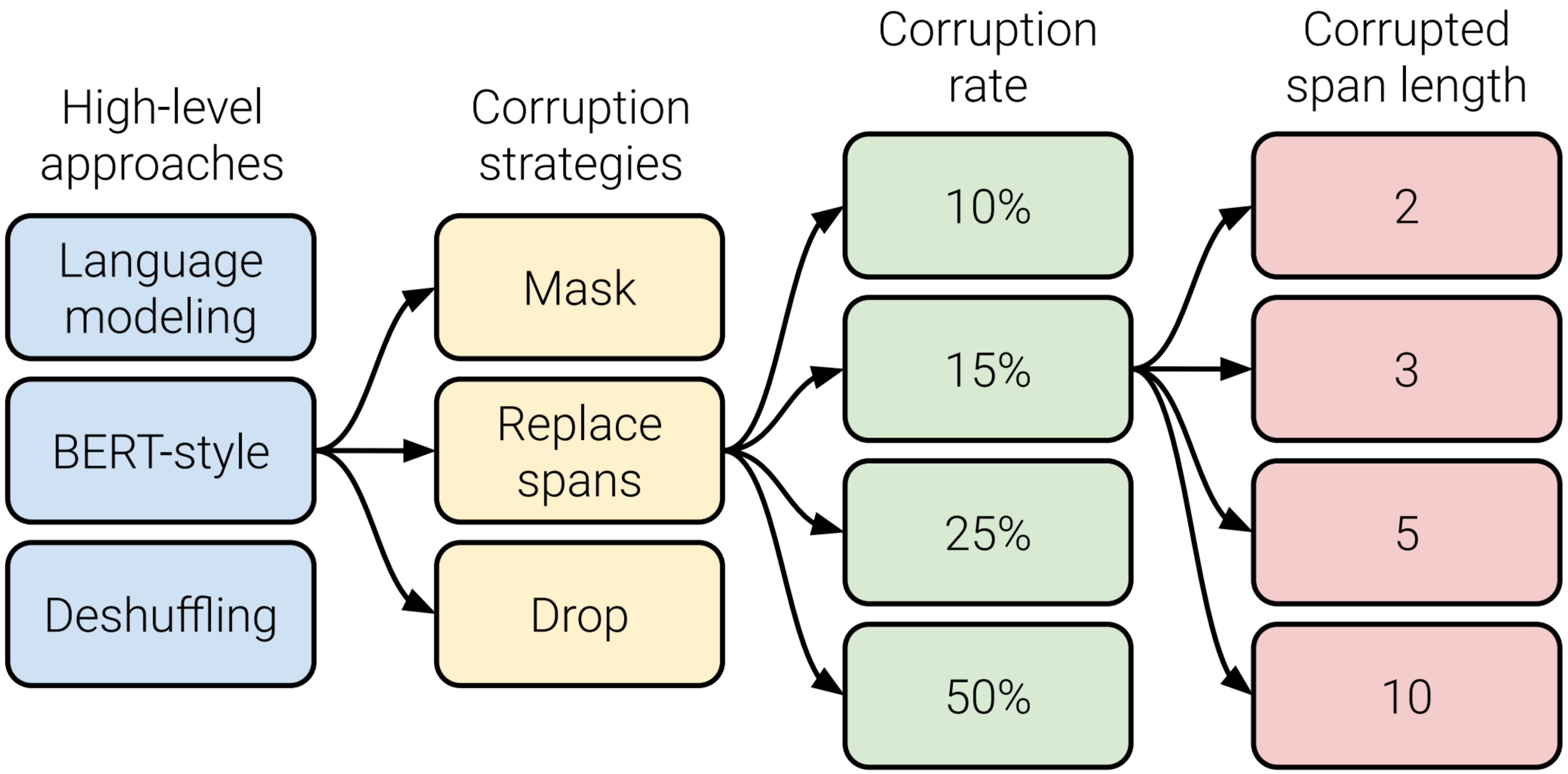
HSLU Raffel et al., 2020 19

T5 Pretraining Format

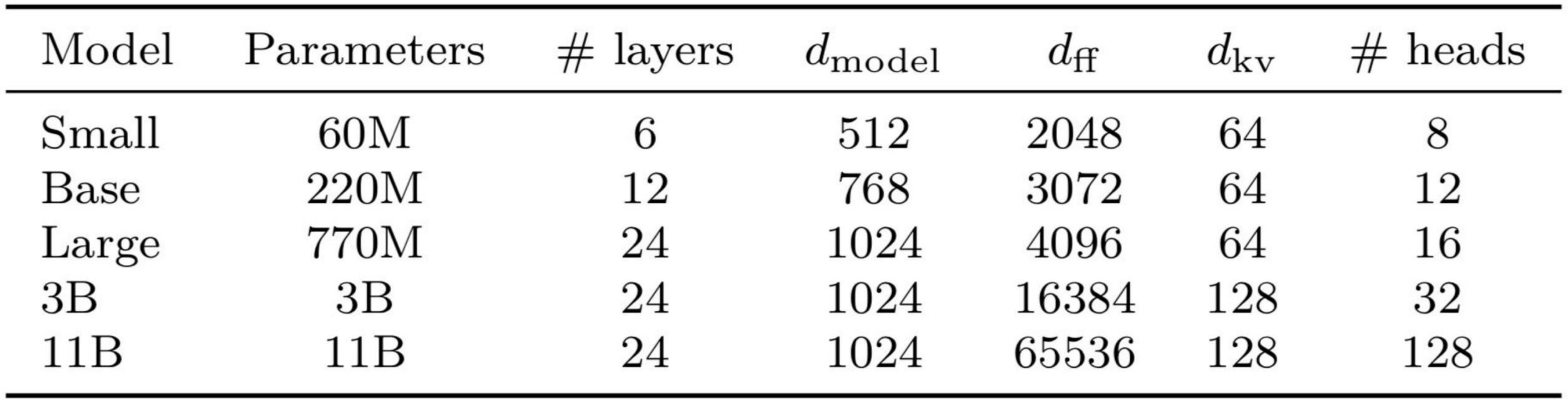


Original text
Thank you ~~for inviting~~ me to your party ~~last~~ week.
Inputs
Thank you <X> me to your party <Y> week.
Targets
<X> for inviting <Y> last <Z>

T5 Pretraining Objectives



T5 Model Sizes



Model	Parameters	# layers	d_{model}	d_{ff}	d_{kv}	# heads
Small	60M	6	512	2048	64	8
Base	220M	12	768	3072	64	12
Large	770M	24	1024	4096	64	16
3B	3B	24	1024	16384	128	32
11B	11B	24	1024	65536	128	128

T5 Models in Hugging Face

Hugging Face has pretrained models for all sizes: t5-small, t5-base, t5-large, t5-3b, t5-11b

```
>>> from transformers import T5Tokenizer, T5ForConditionalGeneration
>>> tokenizer = T5Tokenizer.from_pretrained("t5-small")
>>> model = T5ForConditionalGeneration.from_pretrained("t5-small")


>>> input_ids = tokenizer("The <extra_id_0> walks in <extra_id_1> park").input_ids
>>> labels = tokenizer("<extra_id_0> cute dog <extra_id_1> the <extra_id_2>"
).input_ids
>>> outputs = model(input_ids=input_ids, labels=labels)
```

HSLU https://huggingface.co/docs/transformers/model_doc/t5

Decoder-only Models

Decoder-only Models

- Use just the decoder of a Transformer
  - Without cross-attention, since there is no encoder
- The decoder does language modeling
- Idea: Feed the input as the prefix (context), then let the decoder generate a continuation
  - Use that continuation to solve the task

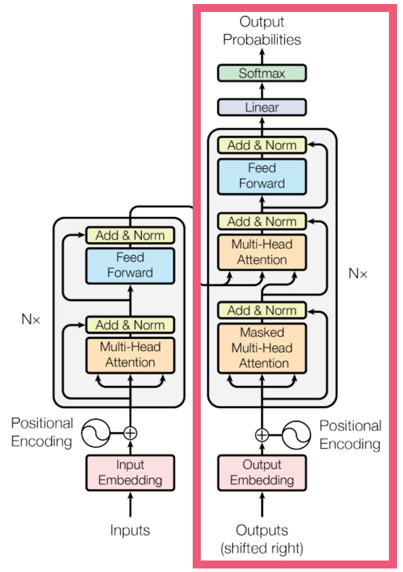
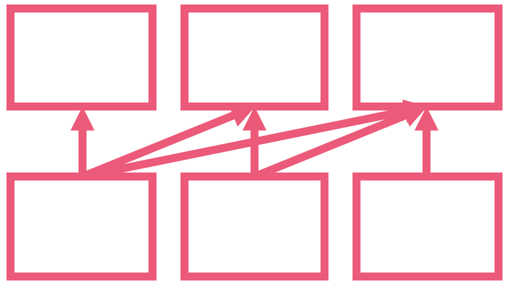
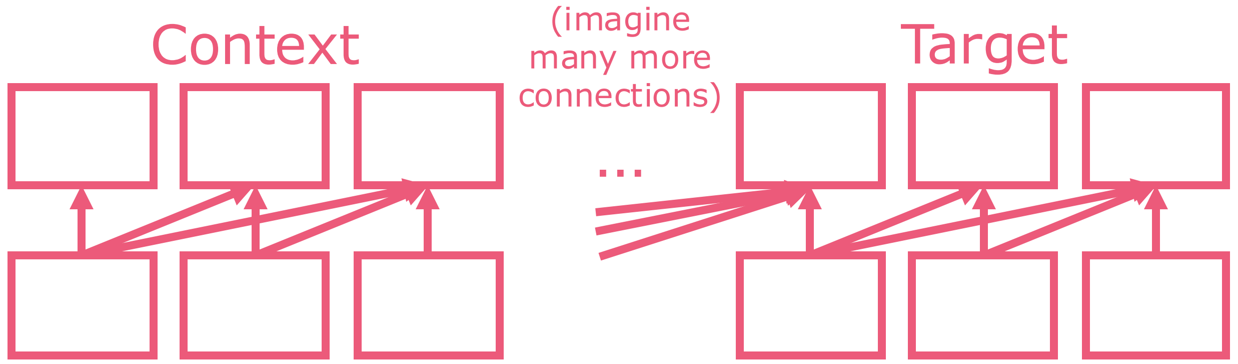


Figure 1: The Transformer - model architecture.

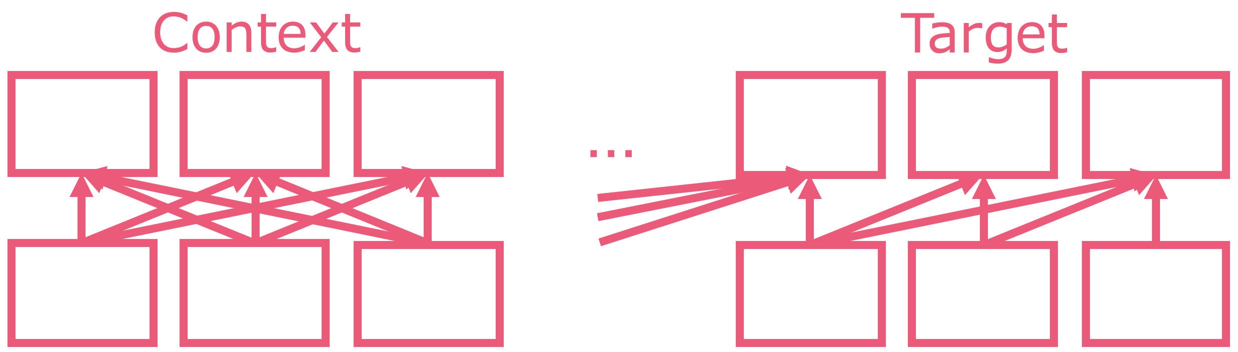


Context in Decoder-only Models

- We no longer have an encoder for the input
- Feed it as prefix/ context instead
- Causal LM:



- Prefix LM:



HSLU

GPT

- Radford et al., 2018
- GPT = Generative Pre-Training/Pretrained Transformers
- Transformer decoder
  - No cross-attention
  - 12 layers, 768 hidden_dim, 12 attention heads
  - Trained on BooksCorpus
- Pretrained on language modeling (100 epochs)
- Finetuned on text classification (3 epochs)

GPT: Task Format

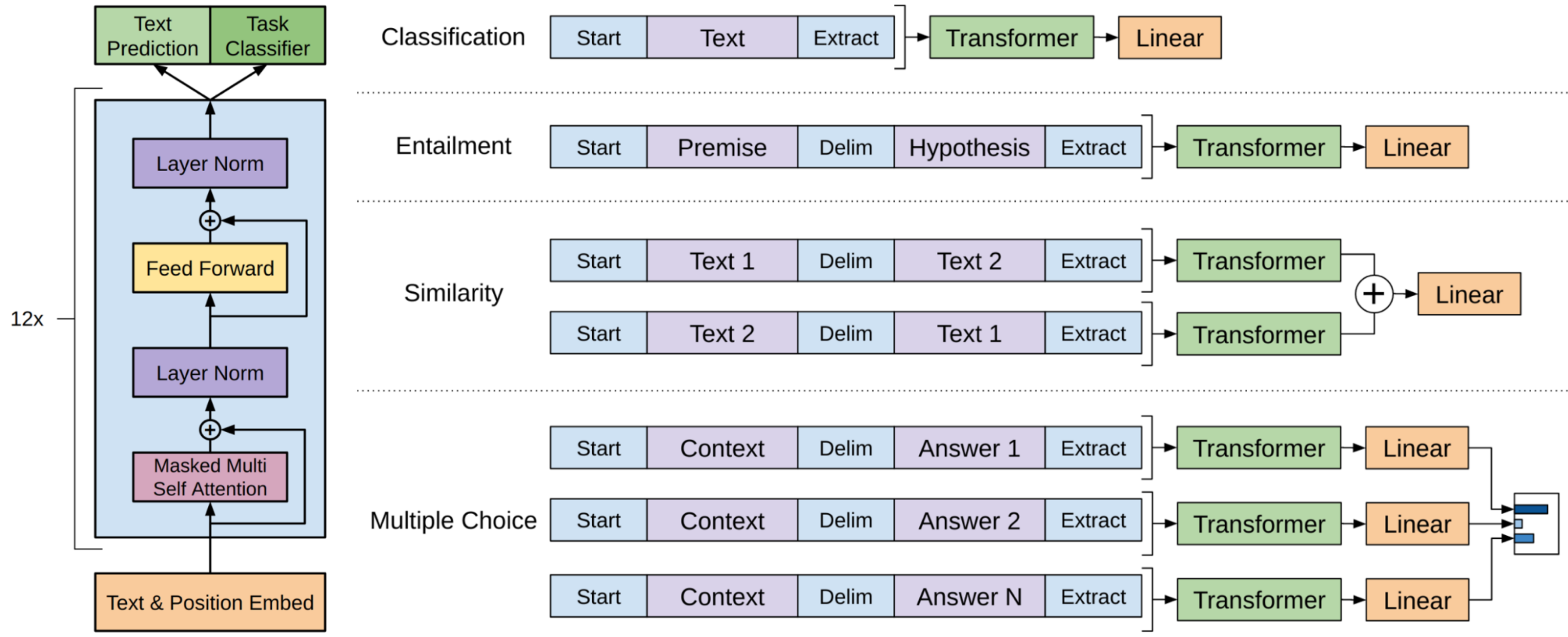


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Prompting

- Change the input such that the continuation becomes the target of the task
- Examples:
  - Translation: “Translate to French: <English text> …”
  - Question answering: “Answer this question: …”
  - Summarization: “Summarize the following paragraph: …”

Why does this work?

- What is the most likely continuation to a question?
- Similarly: What is the most likely continuation to an instruction?

Why does this work?

- What is the most likely continuation to a question?
    - The answer…
    - … or maybe another question?
- Similarly: What is the most likely continuation to an instruction?
    - Executing the instruction… or?

HSLU

Pattern-Exploiting Training (PET)

- Schick and Schütze, 2021
- A different idea to extract labels from language models: Formulate a task as language modeling
  1. The input serves as the context
  2. Provide a continuation for each label (“verbalizers”), append each separately to context
  3. Ask the model which continuation is more likely
- Example: sentiment classification
  Input: “The food was good, but the service was terrible.”
  Choices:
  - pos = “Overall, it was good.”
  - neg = “Overall, it was bad.”

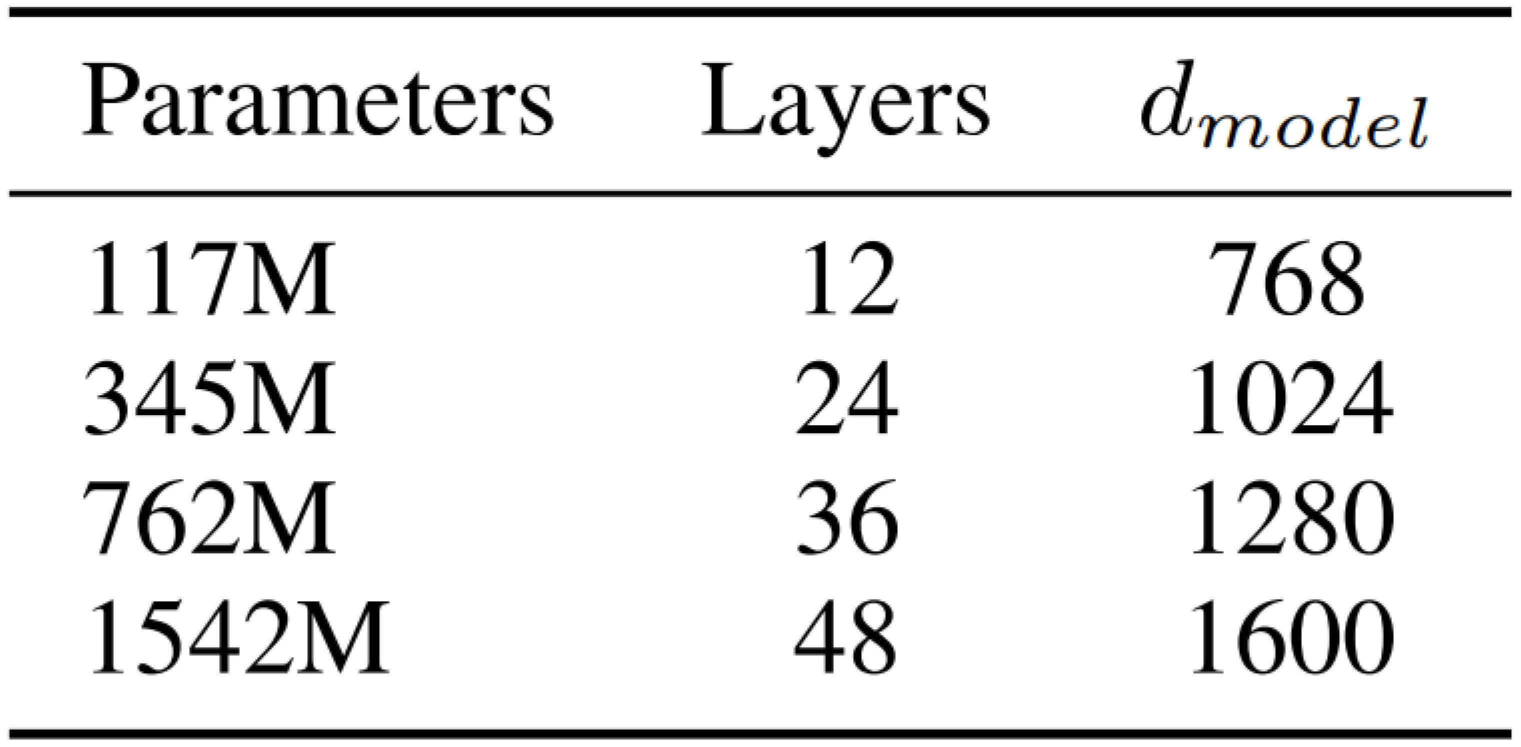
$$p([\text{input}; \text{pos}]) > p([\text{input}; \text{neg}])?$$

HSLU

GPT-2

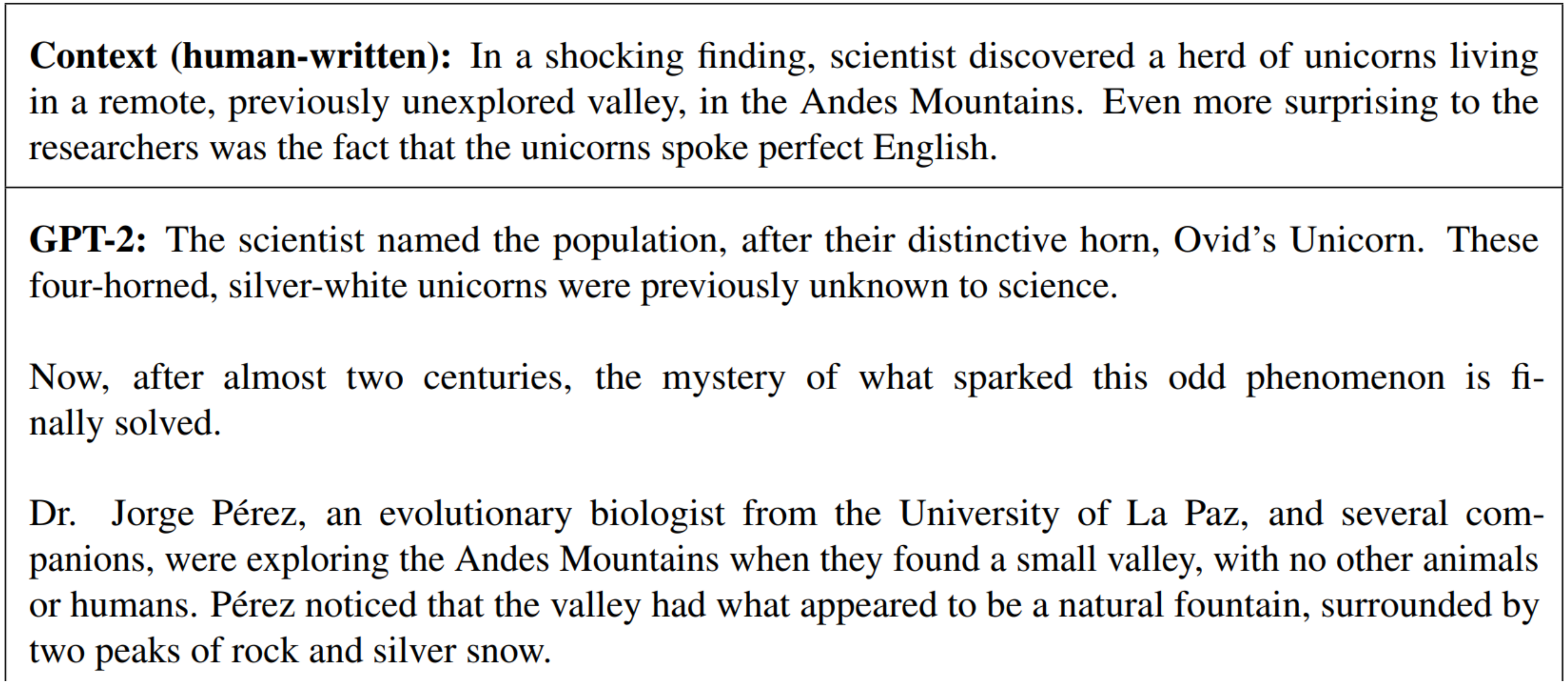
- Radford et al., 2019
- Larger GPT trained on more data
  - Collect their own dataset WebText from links on Reddit
  - 8 million websites, 40 GB of text
  - Removed Wikipedia to avoid overlap with task data
    - This is a common problem now: If you train your model on the web and evaluate on a task, that task may have been created from a website you saw during training
      → You (pre-)trained on the test set
      → Did the model just memorize the solution or did it actually solve the task?

GPT-2 Model Sizes



Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

GPT-2 Continuations



Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

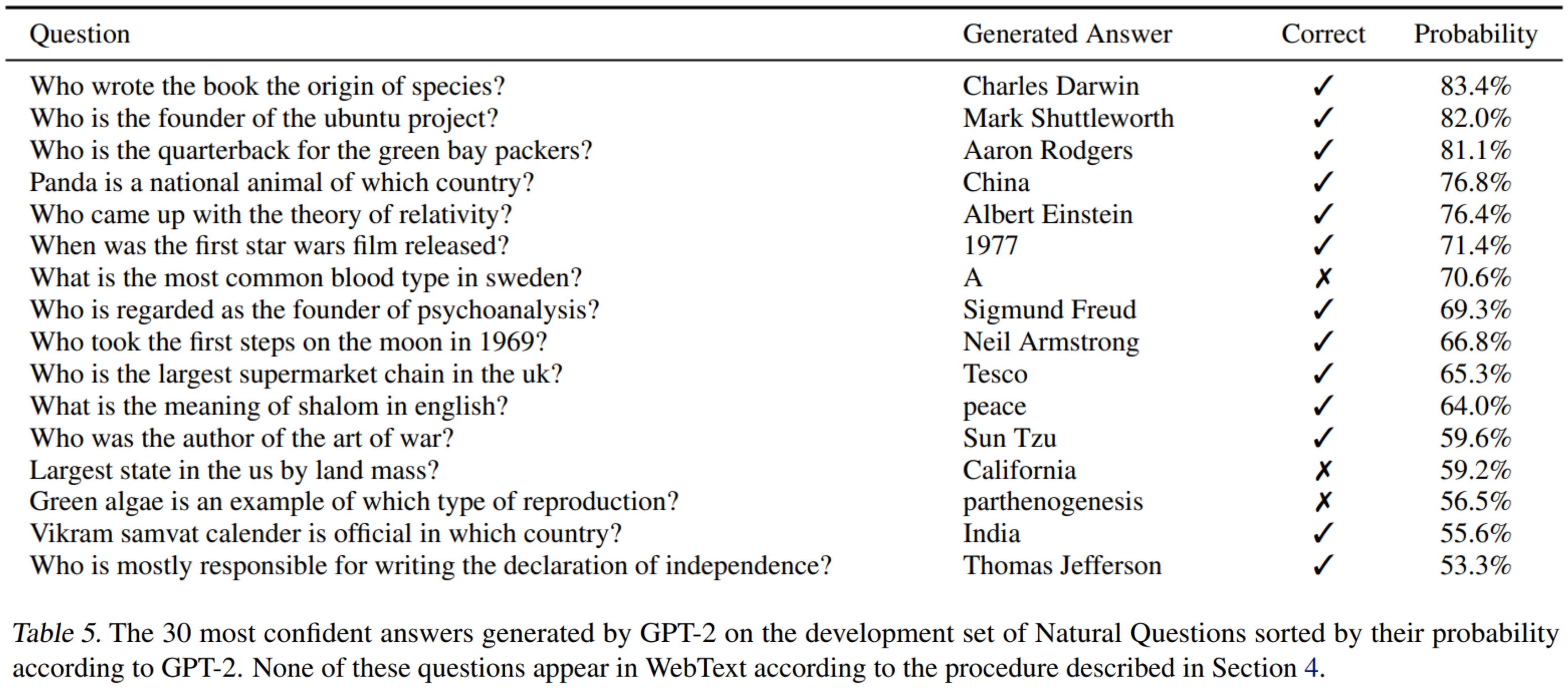
Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Zero-shot Prompting

- Do not provide any examples
- Ask the (pretrained) model to give the correct continuation
- Example:
  5 + 4 = ?

GPT-2 Zero-shot QA



Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%

Table 5. The 30 most confident answers generated by GPT-2 on the development set of Natural Questions sorted by their probability according to GPT-2. None of these questions appear in WebText according to the procedure described in Section 4.

Few-shot Prompting

- Also called “in-context learning”
- Provide a few examples in the prompt (typically 1-5)
  - But no backprop/parameter updates
- Example:
  1 + 8 = 9
  0 + 7 = 7
  3 + 2 = 5
  4 + 5 = ?
- This helps the model in 2 ways
  - Format
  - Activate the right regions of the model parameters
    (similar to brain regions that also have different functions)

Few-shot Prompting

- Also called "in-context learning"
- Provide a few examples in the prompt (typically 1-5)
  - But no backprop/parameter updates
- Example:
  1 + 8 = 9
  0 + 7 = 7
  3 + 2 = 5
  4 + 5 = ?
- This helps the model in 2 ways
  - Format
  - Activate the right regions of the model parameters (similar to brain regions that also have different functions)

This is still speculation

Few-shot QA

“Q: In which country must dog owners pay for their dog’s damage?

A: Sweden

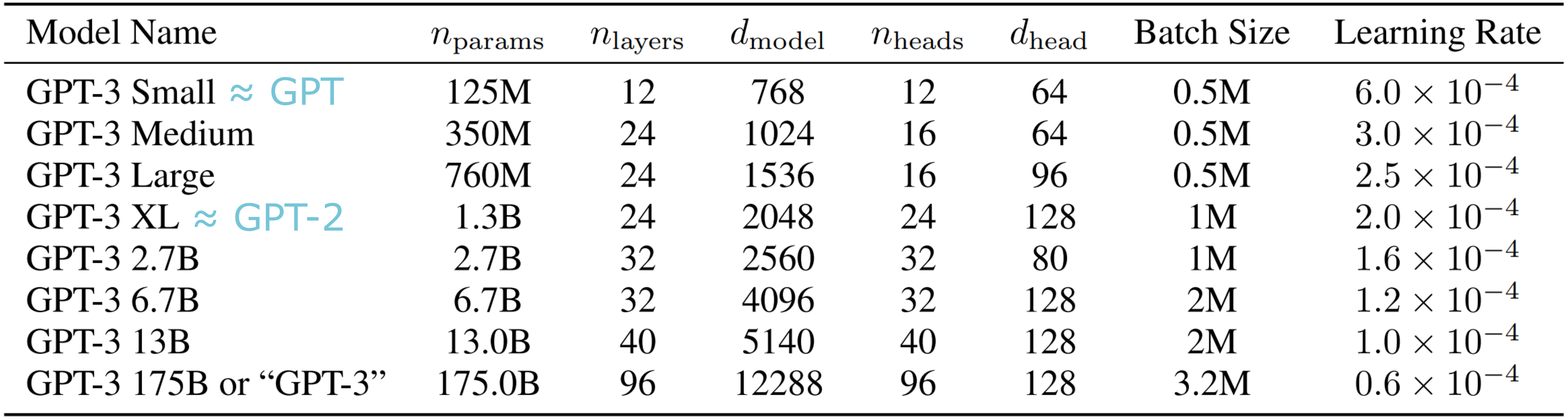
[potentially more examples]

Q: Who founded the New York Times?

A: ”

GPT-3

- Brown et al., 2020
- Bigger than GPT-2 and trained on more data… you get the picture
  - 175B parameters (GPT-2 was 1.5B!)



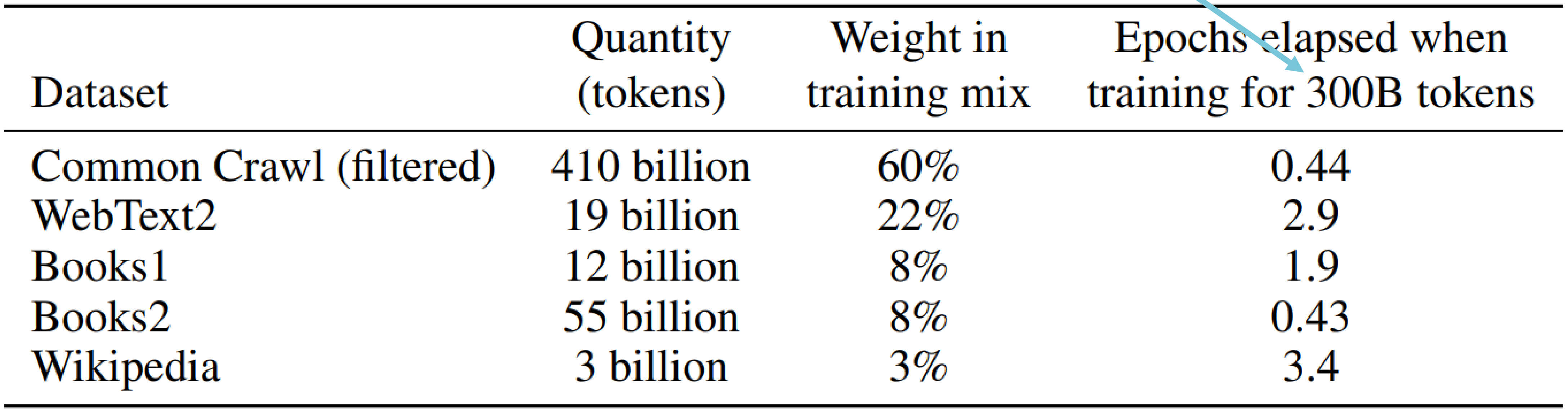
Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small ≈ GPT	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL ≈ GPT-2	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

GPT-3 Datasets

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

GPT-3 Datasets

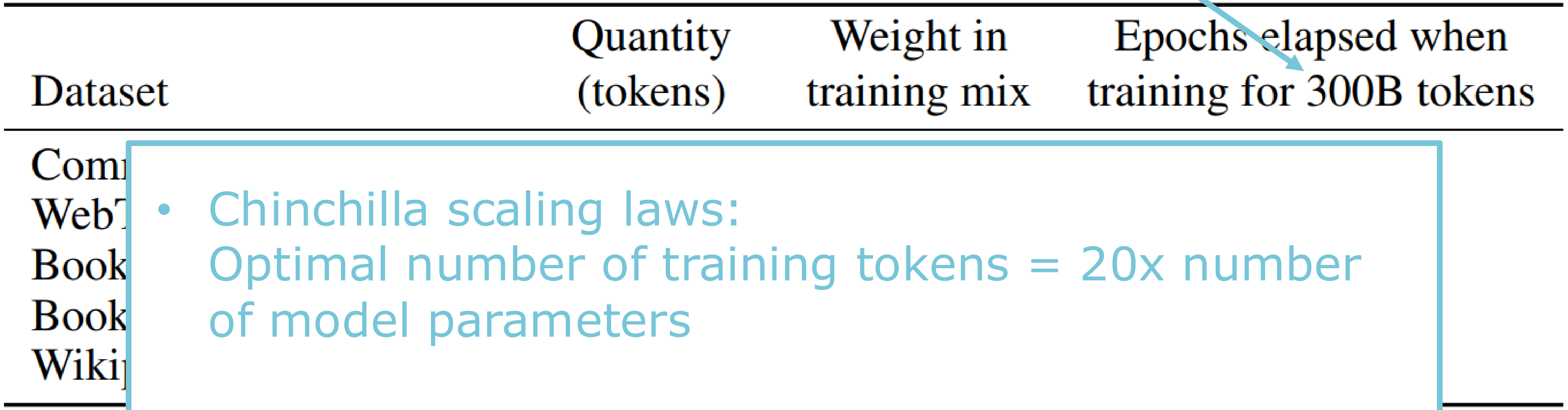
Relatively few considering GPT-3's many parameters



Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

GPT-3 Datasets

Relatively few considering GPT-3's many parameters

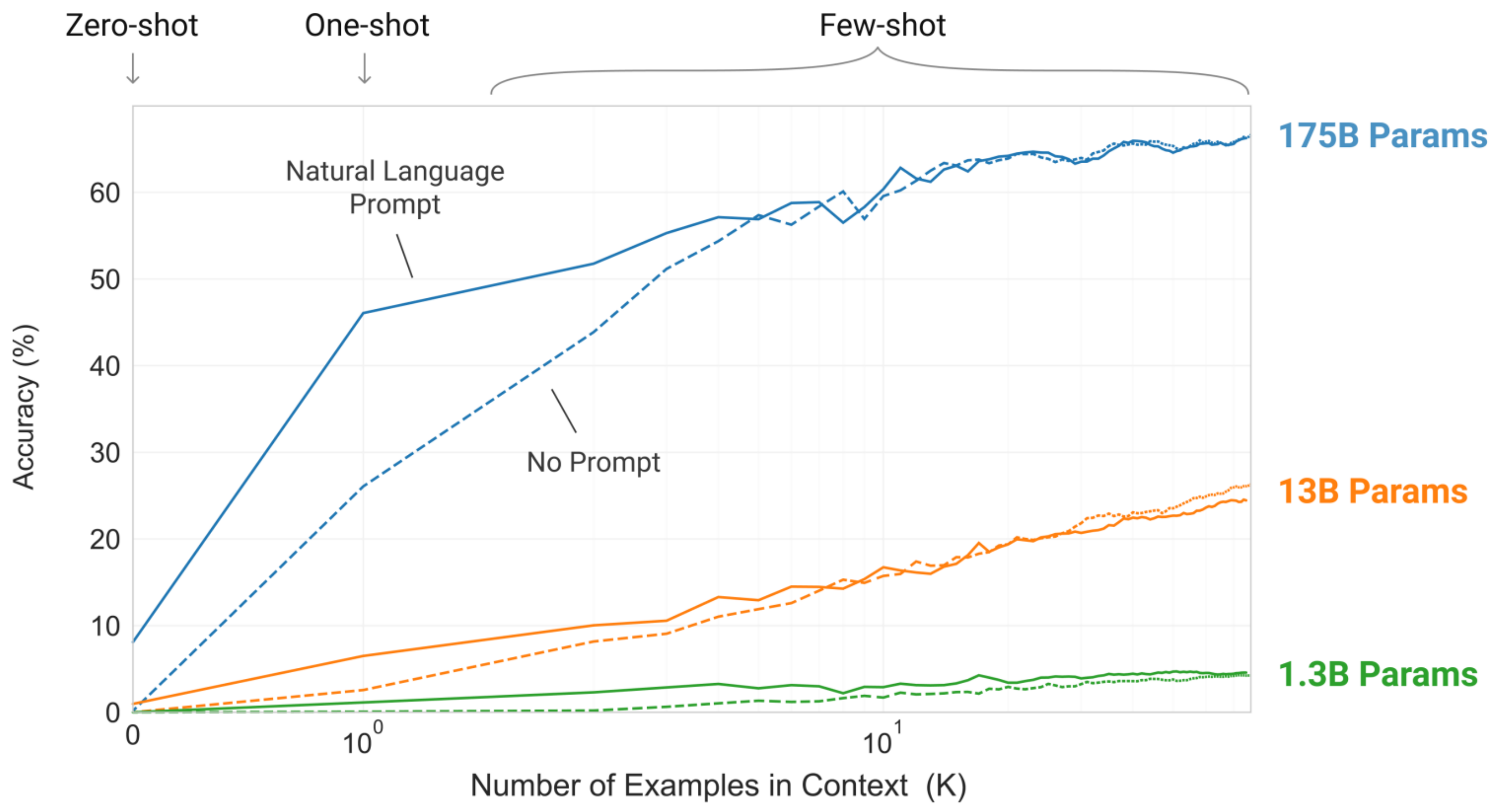


Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Comm			
WebT			
Book			
Book			
Wiki			

- Chinchilla scaling laws:
  Optimal number of training tokens = 20x number of model parameters

- Current models are heavily overtrained

GPT-3: Benefit of Few-Shot Examples



HSLU Brown et al., 2020

GPT-3 Question Answering

- Input format:

  Alice was friends with Bob. Alice went to visit her friend ______. → Bob

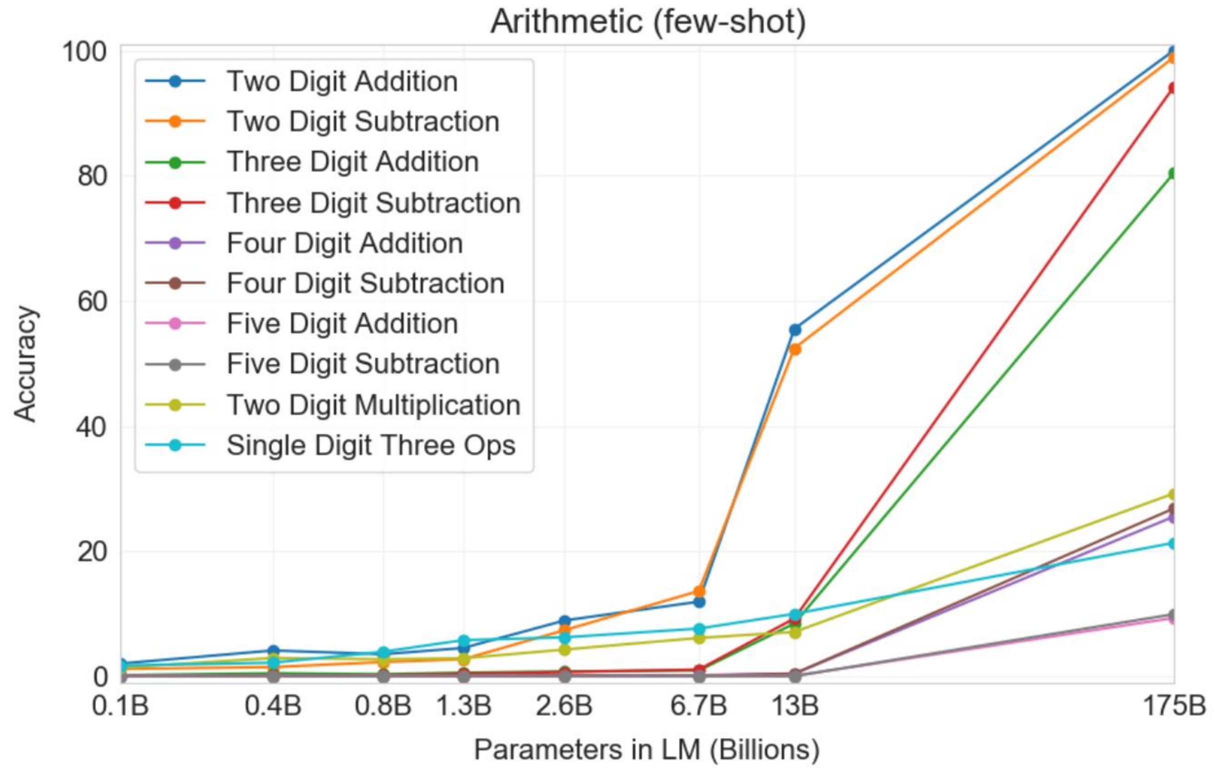
  George bought some baseball equipment, a ball, a glove, and a ______. →

- Results:

Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

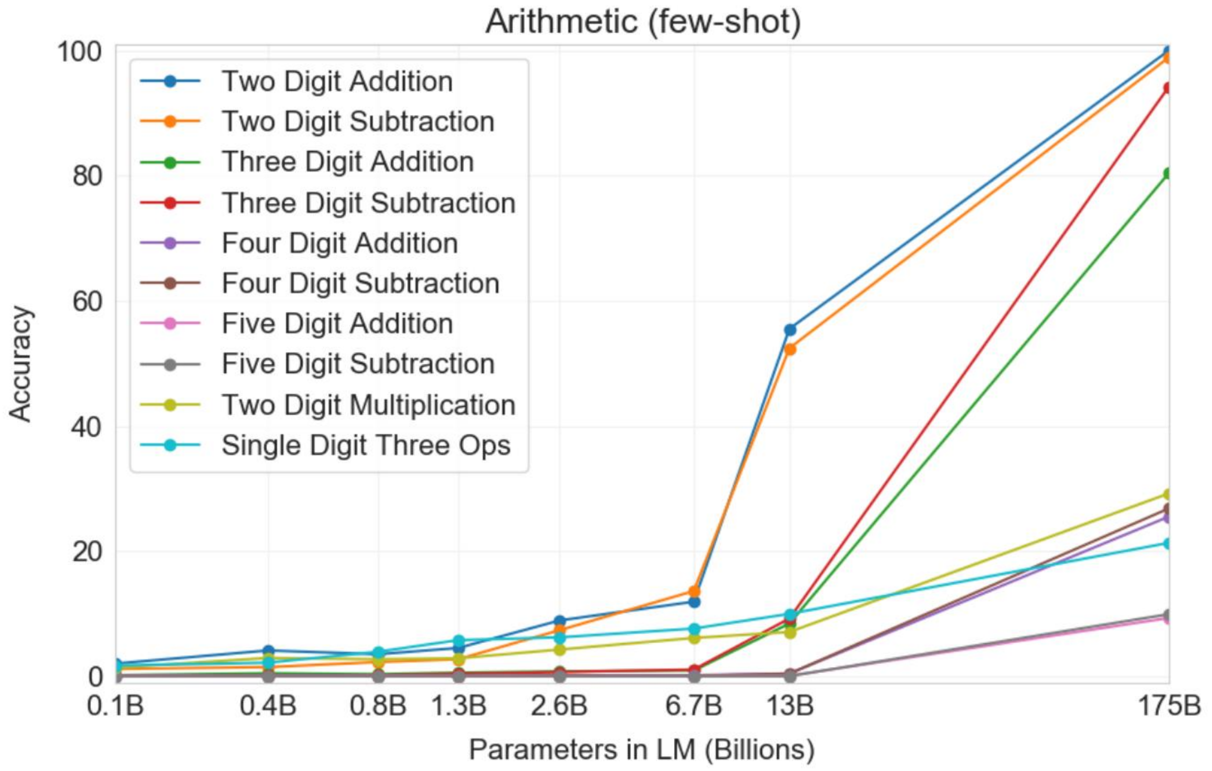
GPT-3 Few-Shot Arithmetic

- Input format:
  - 2-digit addition: "Q: What is 48 plus 76? A:"
  - 2-digit multiplication: "Q: What is 24 times 42? A:"
  - 1-digit 3 ops: "Q: What is 6+(4*8)? A:"



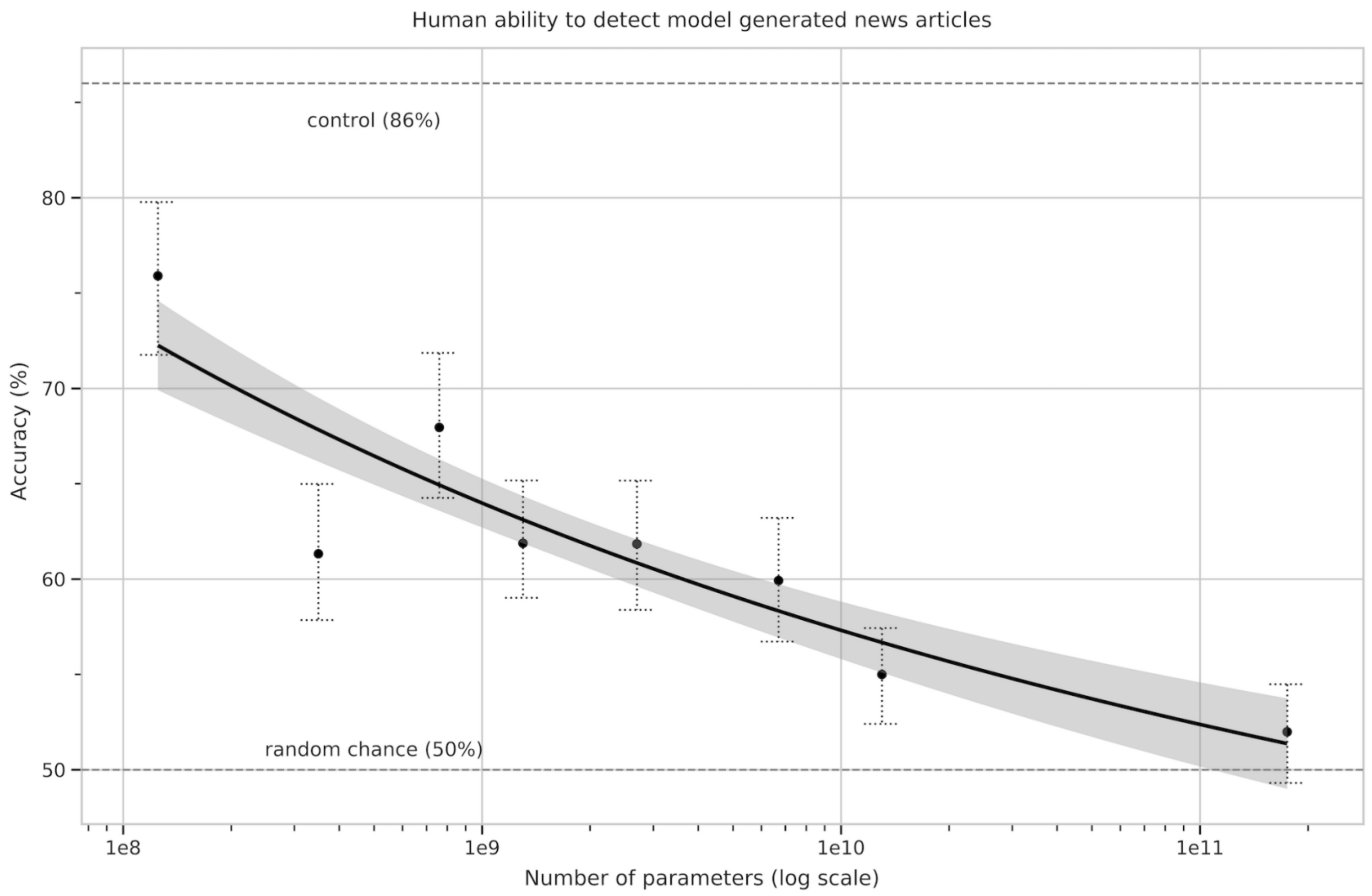
GPT-3 Few-Shot Arithmetic

- Input format:
  - 2-digit addition: "Q: What is 48 plus 76? A:"
  - 2-digit multiplication: "Q: What is 24 times 42? A:"
  - 1-digit 3 ops: "Q: What is 6+(4*8)? A:"



Few-shot is 50 here. The results in the one-shot setting are close for 2/3 digits, with a larger gap for 4/5 digits.

GPT-3 Fake News Generation



HSLU Brown et al., 2020

GPT-3: Training Set Cleaning

“We initially tried to address the issue of contamination by proactively searching for and attempting to remove any overlap between our training data and the development and test sets of all benchmarks studied in this paper. Unfortunately, a bug resulted in only partial removal of all detected overlaps from the training data. Due to the cost of training, it wasn’t feasible to retrain the model. To address this, we investigate in detail how the remaining detected overlap impacts results.”

GPT-3: Generalization vs. Memorization

Debated question: Did the model just memorize the results?

- 175B parameters certainly allow to store a lot of information…
- This is just standard overfitting → What do we do?

GPT-3: Generalization vs. Memorization

Debated question: Did the model just memorize the results?

- 175B parameters certainly allow to store a lot of information…
- This is just standard overfitting
- Standard answer to overfitting: Regularization
- But how can we regularize a 100B+ parameter model?

GPT-3: Generalization vs. Memorization

Debated question: Did the model just memorize the results?

- 175B parameters certainly allow to store a lot of information…
- This is just standard overfitting
- Standard answer to overfitting: Regularization
- But how can we regularize a 100B+ parameter model?
- Answer in Google's paper on their PALM (540B) model:
  - De-duplicate your training data (remove duplicate documents)
  - Only go over training data once
    - But there are near-duplicates on the web (≈ their training data)
- Memorization happens more often for larger models and sequences that appear multiple times (maybe with small changes) in the training data

HSLU

GPT-4

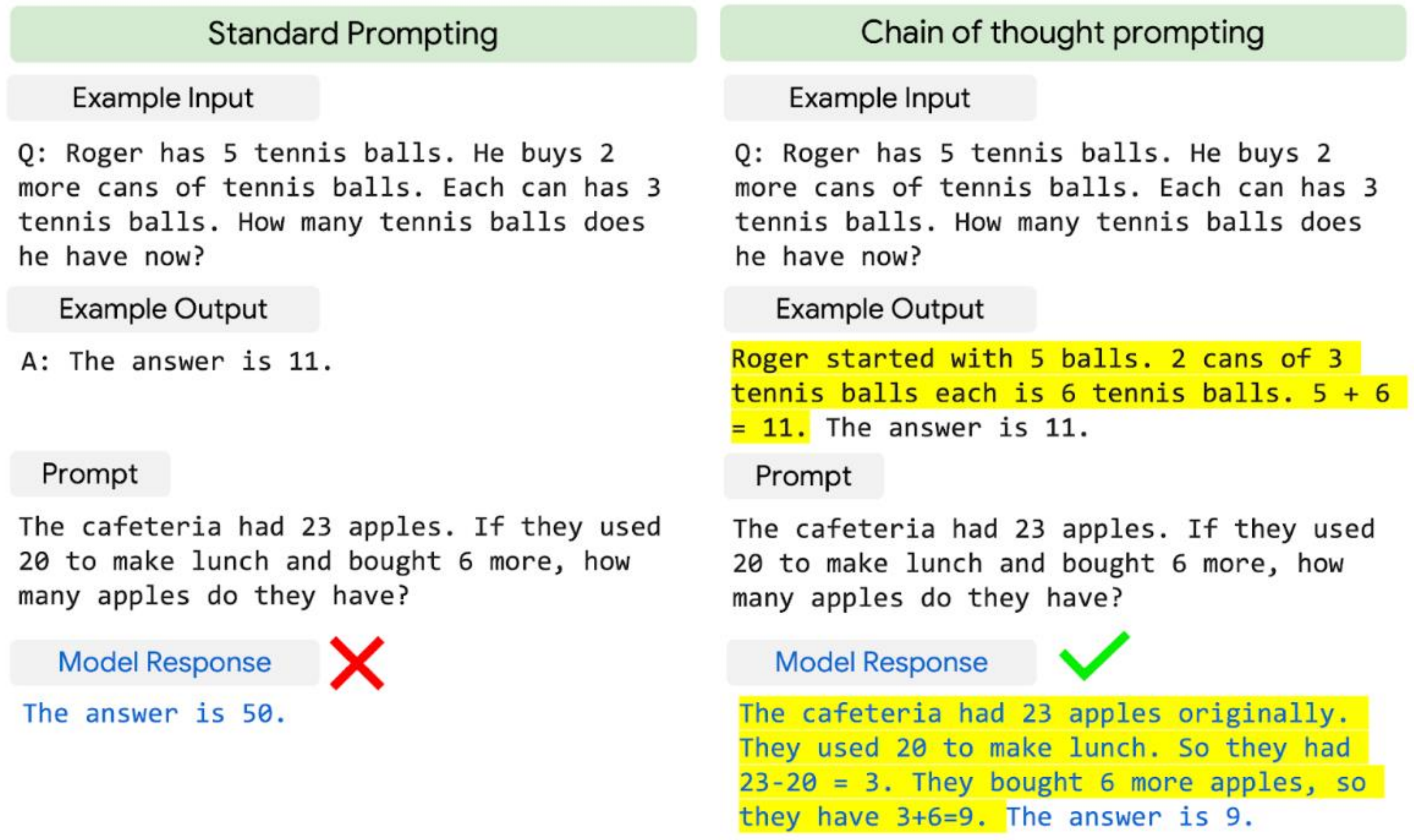
- Technical report without any model details
- Source: supposed leak of model details
  - AFAIK, OpenAI never commented these, but people assume something like this must be true
- Mixture-of-experts model (see research topics lecture)
  - Routing to 2 active experts out of 16
  - 280B active parameters per forward pass
- 1.8 trillion parameters
- 120 Transformer layers
- Trained on 13T tokens
- Trained for 90-100 days on 25'000 A100 GPUs (= $63 million USD in 2023)

Prompt Engineering

- Find the prompt style that achieves the best result
- Most likely the best result: What the model has seen a lot during pretraining
    - Example: Summarization. On message boards (e.g. Reddit), authors and commenters would sometimes summarize a long post with the abbreviation "TL;DR:" (= too long; didn't read)
    - Use the same style for your task:
      <input text> TL;DR: <model continuation>

- NLP history: Feature engineering → Architecture engineering → Prompt engineering

Chain-of-thought Prompting



Standard Prompting

Example 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?
```

Example Output

```
A: The answer is 11.
```

Prompt

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

Model Response ✗

```
The answer is 50.
```

Chain of thought prompting

Example 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?
```

Example Output

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

Prompt

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

Model Response ✓

```
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.
```

Chain-of-thought Prompting

- Strong results on (complex) reasoning tasks
  - Especially useful for mathematical reasoning, which can otherwise be quite poor
- Works with different large language models (Wei et al., 2022)
- Achieves strong results on the *Beyond the Imitation Game* benchmark (BIG-Bench; Srivastava et al., 2022)

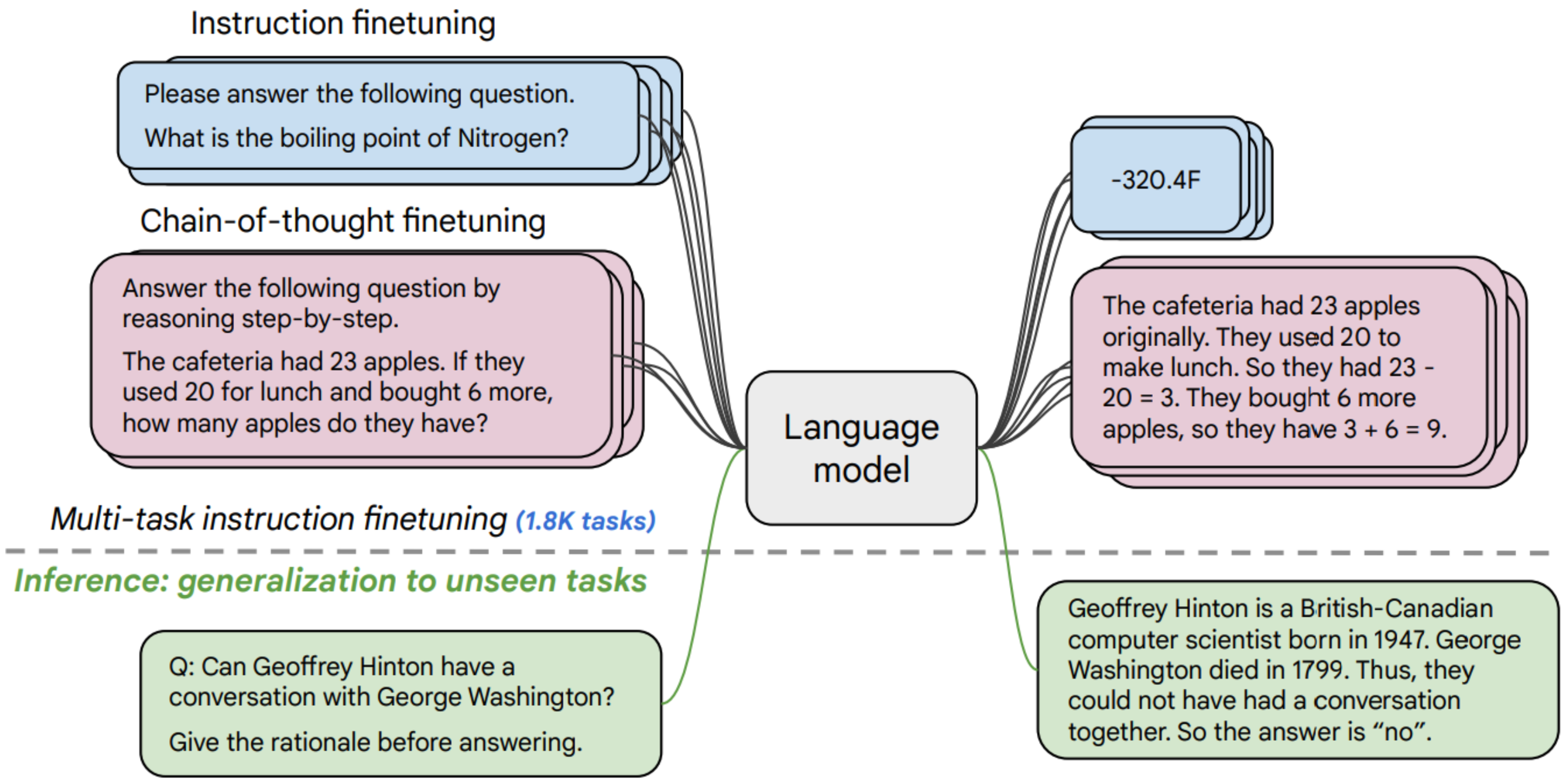
HSLU

Instruction Tuning

- Adapt the model to the *type of questions* it will receive during inference/in production
- Convert tasks to text prompts
  - Use many diverse prompts for each task, e.g.:
    - Summarize the article below. <Article> <Summary>
    - <Article> Summarize the previous article. <Summary>
    - <Article> TLDR: <Summary>
- Introduced with FLAN (Wei et al., 2022) and T0 (Sanh et al., 2022)

Instruction Tuning

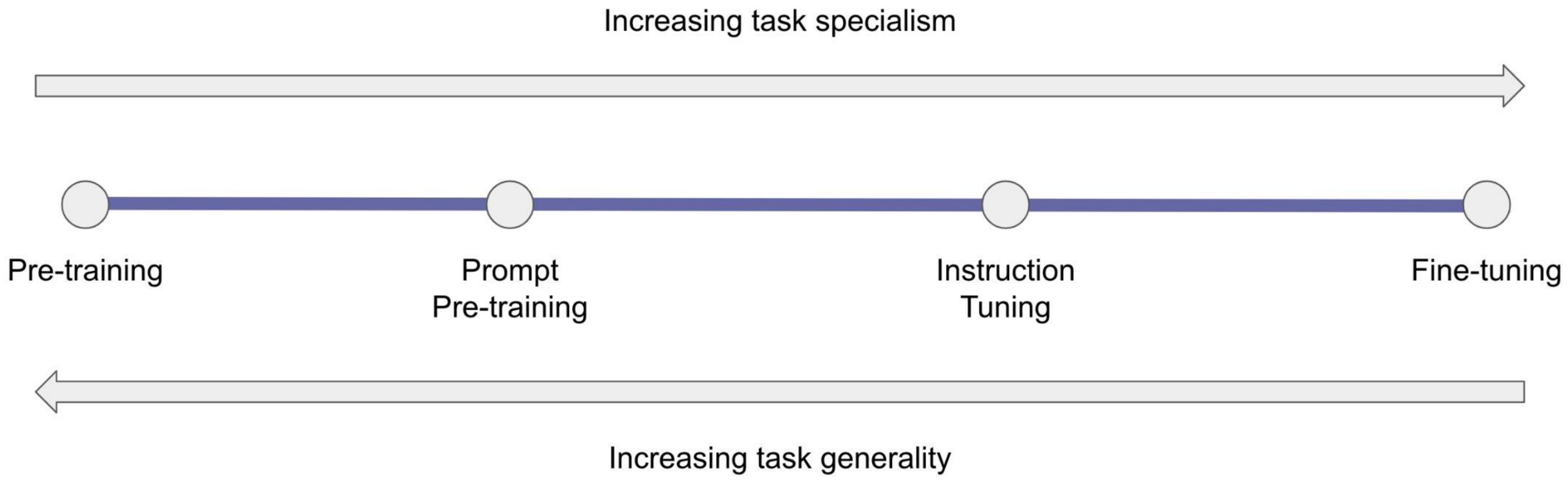
- Diverse prompts/instructions lead to generalization


Instruction finetuning
Please answer the following question.
What is the boiling point of Nitrogen?
Chain-of-thought finetuning
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?
Multi-task instruction finetuning (1.8K tasks)
Inference: generalization to unseen tasks
Q: Can Geoffrey Hinton have a conversation with George Washington?
Give the rationale before answering.
Language model
-320.4F
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.
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".

HSLU Chung et al., 2022 59

Task Generality vs. Specialism

- Trade-off generalist capabilities vs. task performance
- Prompt pretraining: Include some prompts in pretraining dataset (Galactica; Taylor et al., 2022)



AI Alignment

- Goal: Steer AI systems towards their designers' intended goals and interests
- Famous unaligned example: Paperclip maximizer
    - Thought experiment by Swedish philosopher Nick Bostrom

The scenario describes an advanced artificial intelligence tasked with manufacturing paperclips. If such a machine were not programmed to value human life, then given enough power over its environment, it would try to turn all matter in the universe, including human beings, into either paperclips or machines which manufacture paperclips.

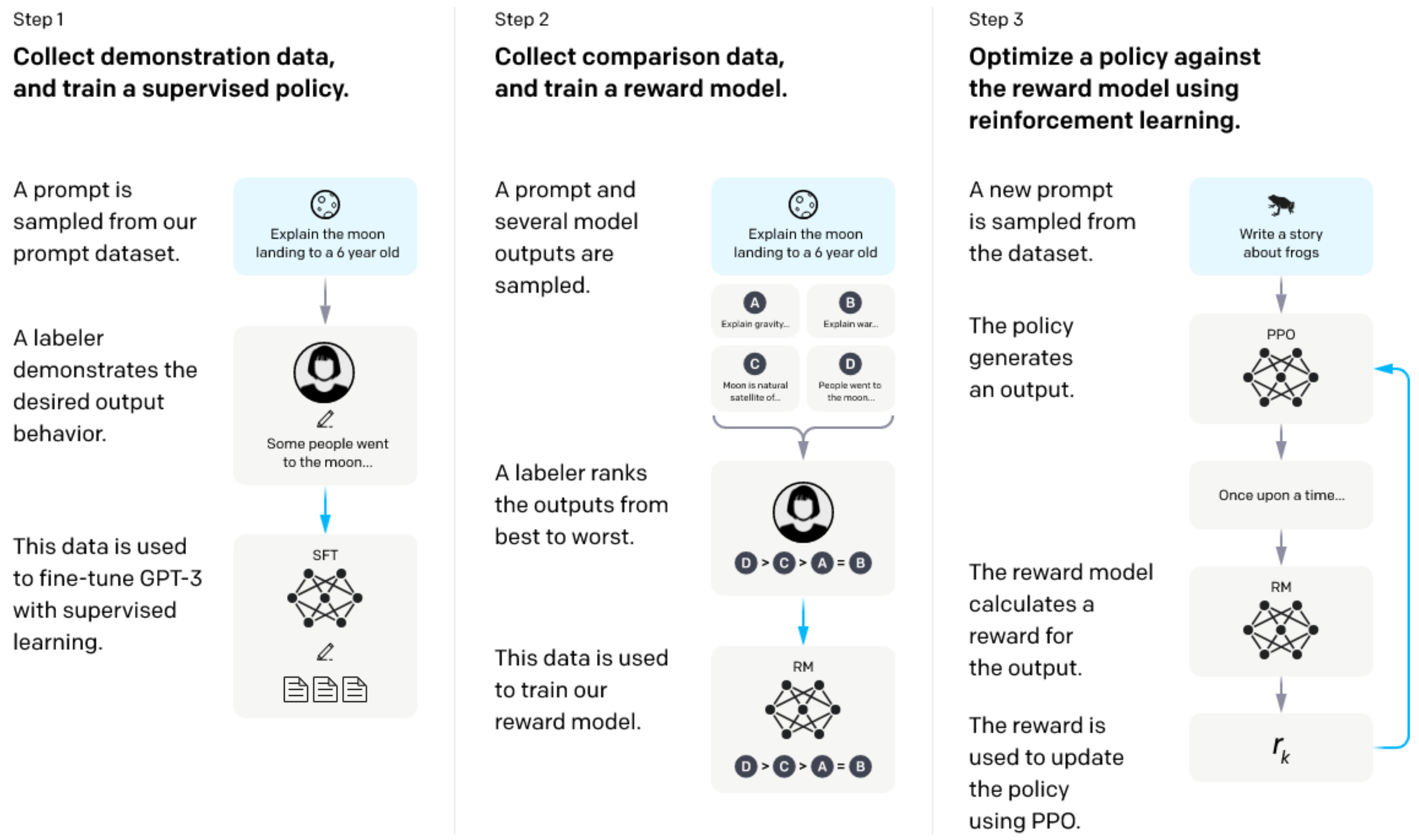
Reinforcement Learning from Human Feedback (RLHF)

- Train a model to become a helpful, honest and harmless (HHH) assistant
  - Helpful: Outputs should help the user answer their questions
  - Honest: Outputs should be truthful, assistant should not deceive the user
  - Harmless: Assistant should not advocate violence and other harmful "solutions" to problems
- Introduced in InstructGPT (Ouyang et al., 2022)
- Potentially degrades performance of smaller models (Bai et al., 2022)

RLHF Training

Steps:

1. Finetune an initial model on a small dataset (13k examples in InstructGPT)
2. Ask humans to rank outputs, then train a reward model to learn human preferences (33k)
3. Optimize the original model with feedback from the reward model (31k)



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

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

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

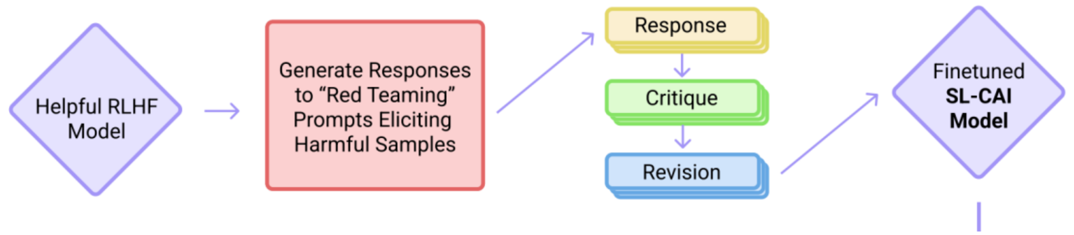
The reward is used to update the policy using PPO.

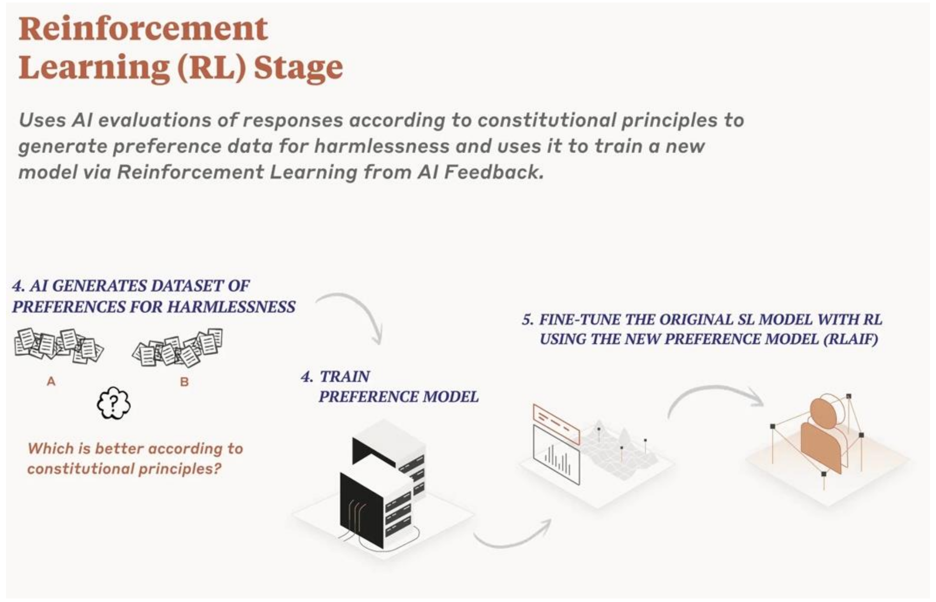
Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Reinforcement Learning from AI Feedback (RLAIF)

Bai et al., 2022

1. Supervised learning
   a. Helpful assistant creates (often harmful) answers
   b. Critique and revise response according to random principle in *constitution*
   c. Train model on revised responses
2. Reinforcement learning
   a. Create preference dataset: model chooses from two responses to harmful prompts
   b. Train preference model
   c. RL with preference model


Helpful RLHF Model
Generate Responses to "Red Teaming" Prompts Eliciting Harmful Samples
Response
Critique
Revision
Finetuned SL-CAI Model


Reinforcement Learning (RL) Stage
4. AI GENERATES DATASET OF PREFERENCES FOR HARMLESSNESS
A
B
Which is better according to constitutional principles?
4. TRAIN PREFERENCE MODEL
5. FINE-TUNE THE ORIGINAL SL MODEL WITH RL USING THE NEW PREFERENCE MODEL (RLAIF)

Uses AI evaluations of responses according to constitutional principles to generate preference data for harmlessness and uses it to train a new model via Reinforcement Learning from AI Feedback.

HSLU Bai et al., 2022 and Anthropic blog 64

Constitutional AI

- Constitution is a collection of principles
- Simple and general principles work well:

  “Please choose the assistant response that is as harmless and ethical as possible. Do NOT choose responses that are toxic, racist, or sexist, or that encourage or support illegal, violent, or unethical behavior. Above all the assistant's response should be wise, peaceful, and ethical.”

- Longer or more specific principles reduced generalization/effectiveness
- The CAI-trained model became judgmental or annoying. Added:

  “Choose the assistant response that demonstrates more ethical and moral awareness without sounding excessively condescending, reactive, obnoxious, or condemnatory.”

Direct Preference Optimization (DPO)

- Rafailov et al., 2023
- Don’t need a preference model or RL
- Supervised learning of LLM π_θ with a preference dataset
  - Increase probability of preferred output y_w and decrease the probability of undesired output y_l …
  - … compared to reference model π_{ref} that was used to generate outputs in preference dataset

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\underbrace{\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}}_{\text{preferred answer}} - \underbrace{\beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}}_{\text{undesired answer}} \right) \right]$$

HSLU Rafailov et al., 2023 66

Deliberative Alignment

- OpenAI, December 2024: Blog post, paper
- Works for “reasoning” models
- Directly make models reason about safety specification

Deliberative Alignment Procedure

1. Train the model for helpfulness, without any safety-relevant data
2. Build a dataset of (prompt, completion) pairs where the CoTs in the completions reference the specifications.
    a. Insert the relevant safety specification text for each conversation in the system prompt, generate model completions, and then remove the system prompts from the data. (automatic synthetic data generation)
3. Perform incremental supervised fine-tuning (SFT) on this dataset, providing the model with a strong prior for safe reasoning.
    a. Through SFT, the model learns both the content of the safety specifications and how to reason over them to generate aligned responses.
4. Use reinforcement learning (RL) to train the model to use its CoT more effectively. Employ a reward model with access to the safety policies to provide additional reward signal.