Foundations: Finetuning, in-context learning, prompting, learning from human feedback

Sebastian Schuster

Seminar “What do language models really understand”?
April 20, 2023

Plan for today

- Masked language models and autoregressive language models
- Fine-tuning models for specific tasks
- In-context learning and prompting
- Learning from human feedback

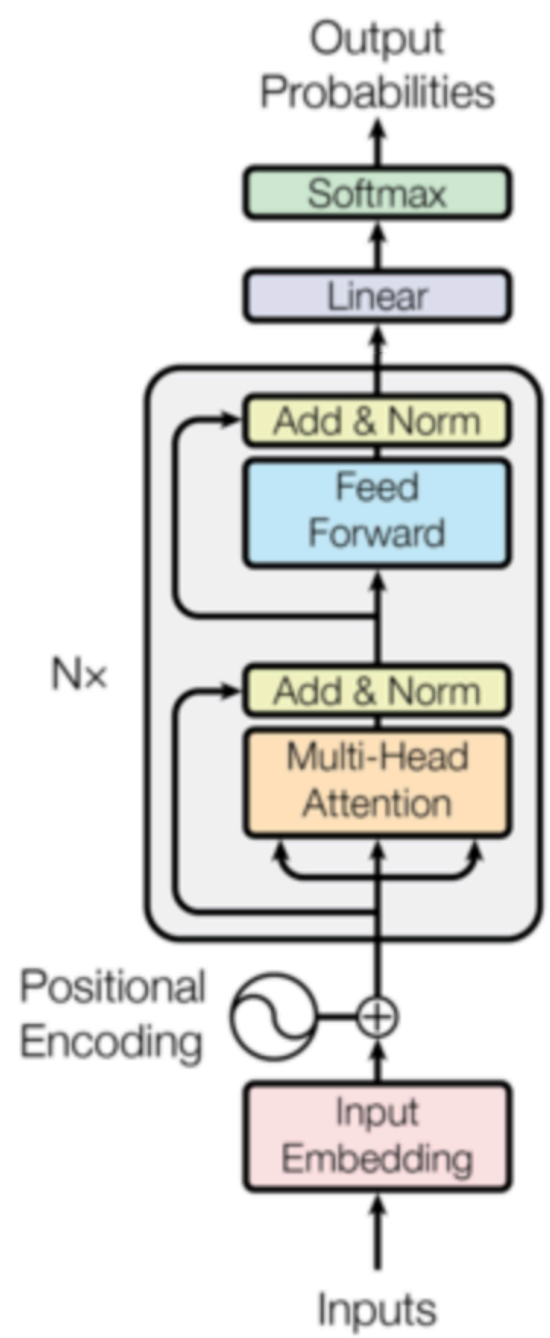
More organizational matters

- Computer Science seminar registration chaos
- Course management system:
  - Computer science CMS: https://cms.sic.saarland/wdlmu23
  - Will be used for submitting questions (private) and announcements
- I will post additional readings and materials on course website (soon!)

Presentation signup

- Early next week (I will finalize the paper list by then)
    - I will ask you for 5 papers that you would like to present
- I will do assignments by mid-next week

Reminder from last time: Transformers

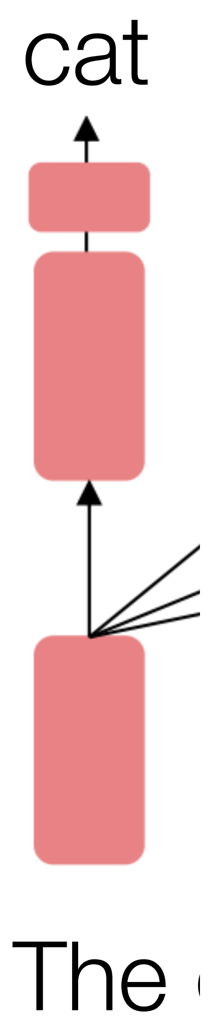


Training objectives

- Next word prediction (*autoregressive LMs*)
  - **Model input:** Previous context up to word w_k
  - **Objective**: Assign high probability to word w_{k+1}
- Masked language modeling (*masked LMs*)
  - **Model input:** Entire context, with about 10% of the words masked or replaced with a random word
  - **Objective:** Assign high probability to original words in the input

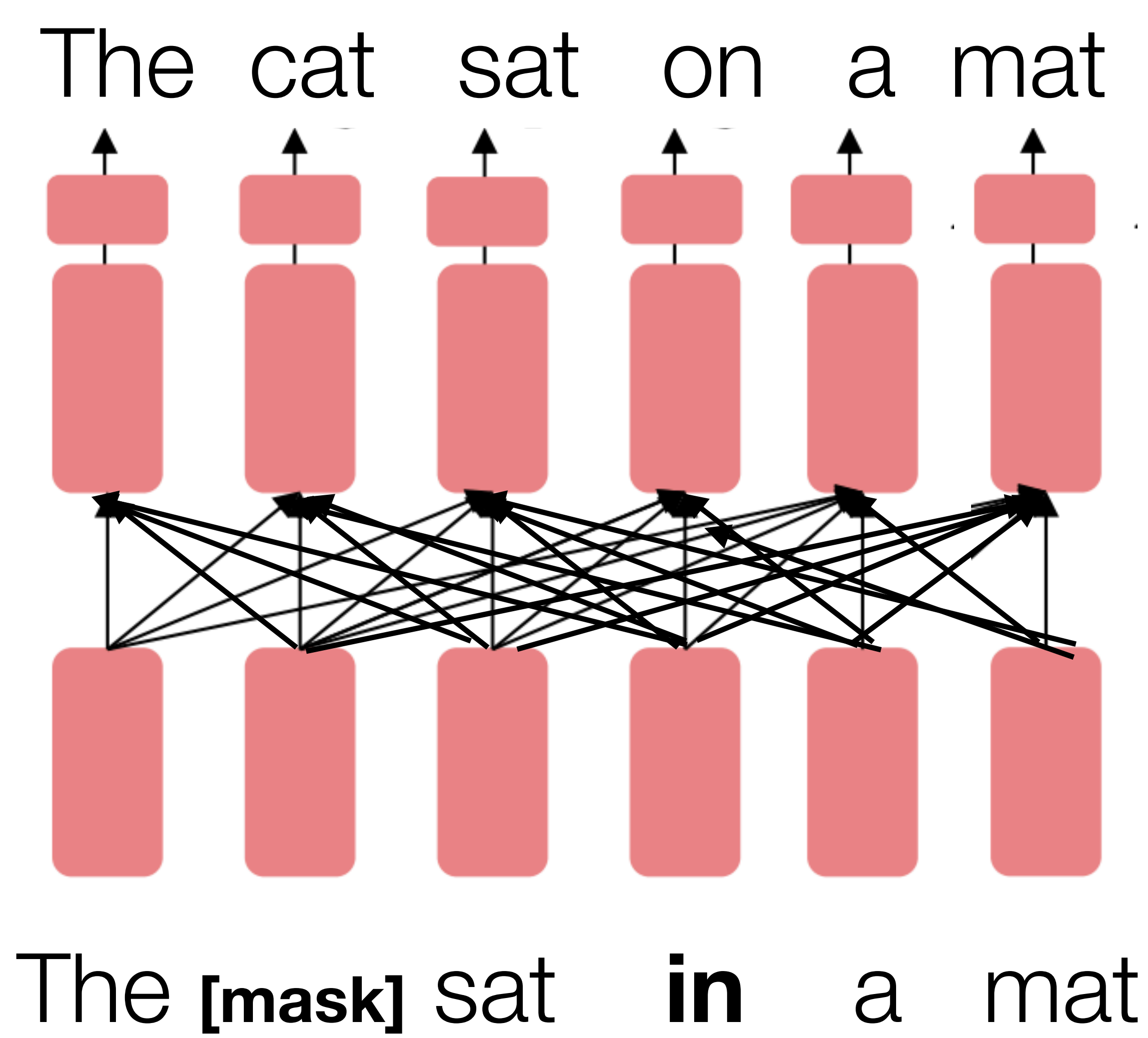
Autoregressive language models

- **Corpus:** The cat sat on a mat

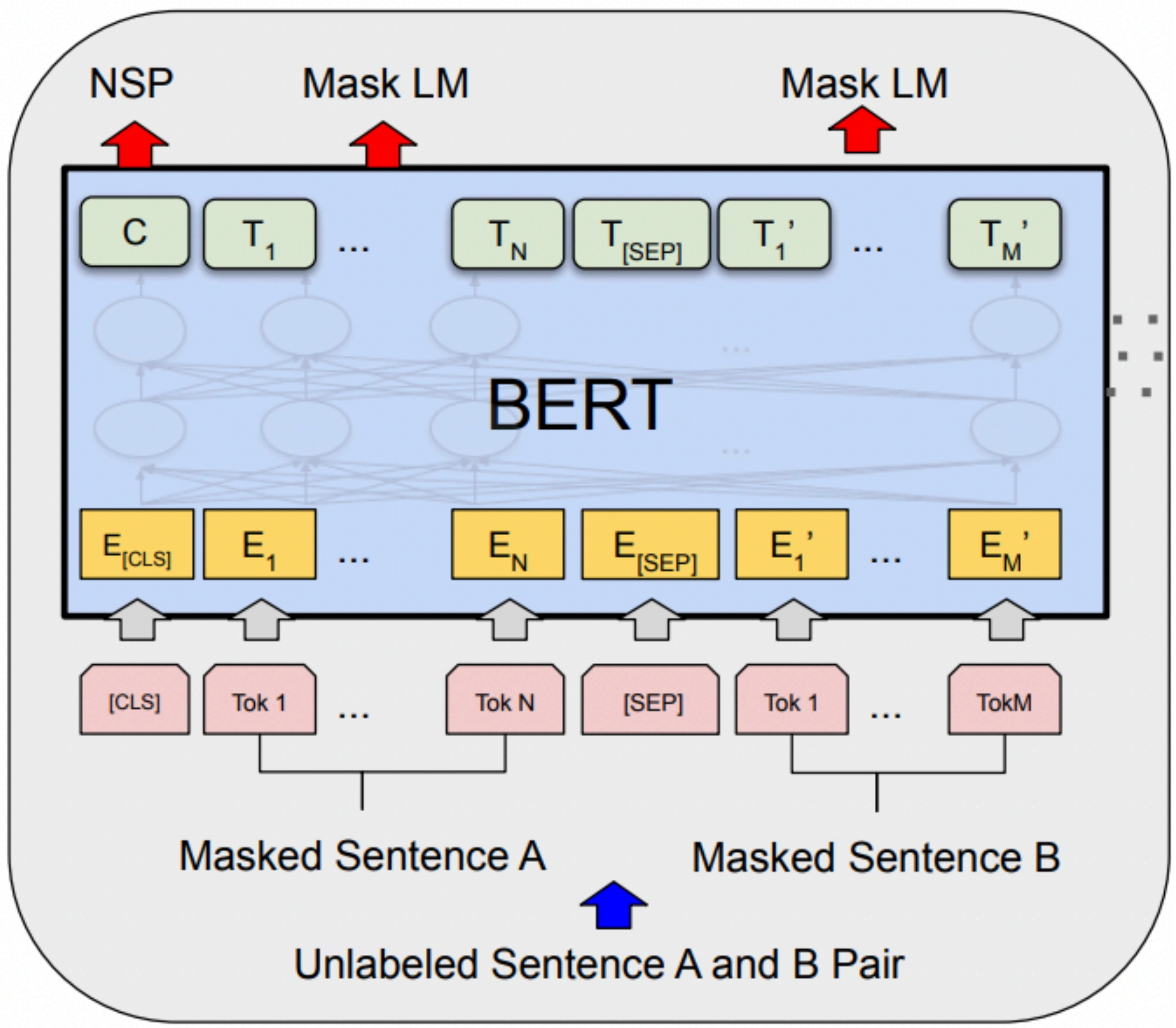


Masked language models

- **Corpus:** The cat sat on a mat



BERT



Two objectives: Masked LM and Next Sentence Prediction (NSP)

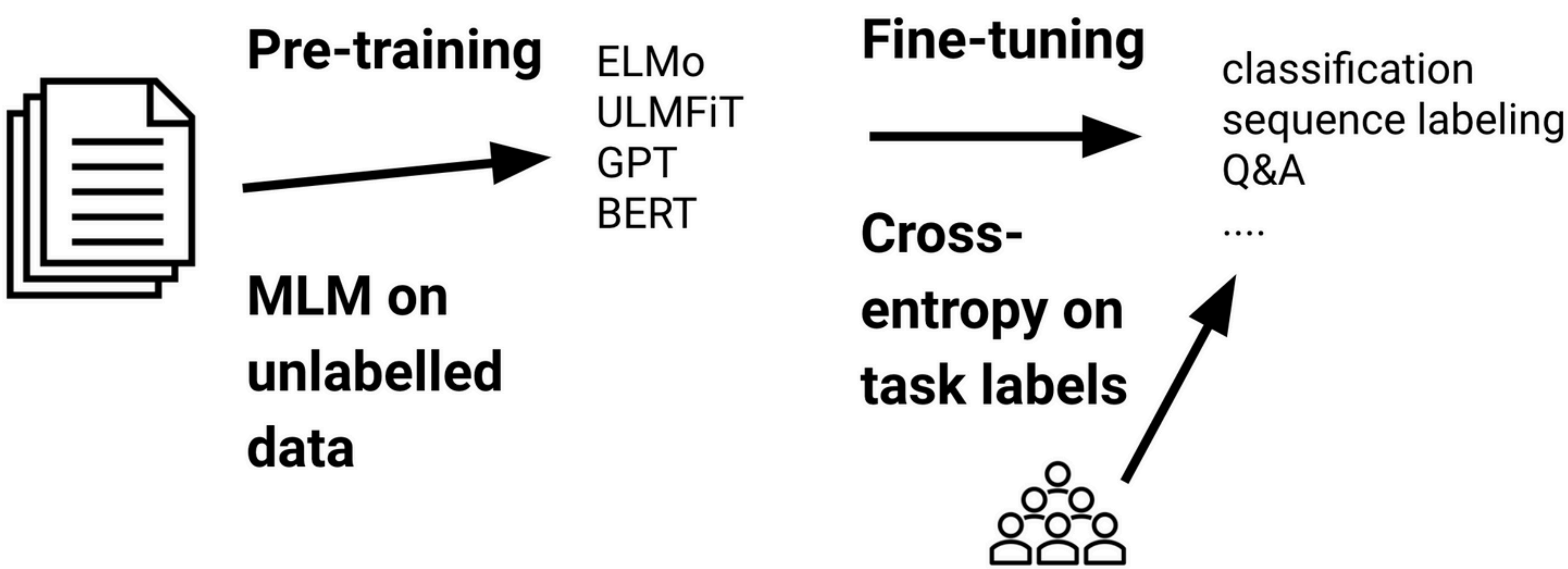
Devlin et al. (2019)

Pretraining-Finetuning Paradigm

Pretraining and finetuning

- **Pretraining** with next word prediction task or masked LM task can be done on **unlabeled text** — no special annotations needed!
- Often done one corpora with billions of words
- The model **learns good representations** (e.g., word vectors) through pretraining
- For a specific task, representations can be **finetuned** on several hundred of thousand examples
  - Pretrained models learn much faster and generalize better than models initialized from scratch!

Pretraining and finetuning



Adapted from https://ruder.io/content/images/2021/02/fine-tuning_methods.png

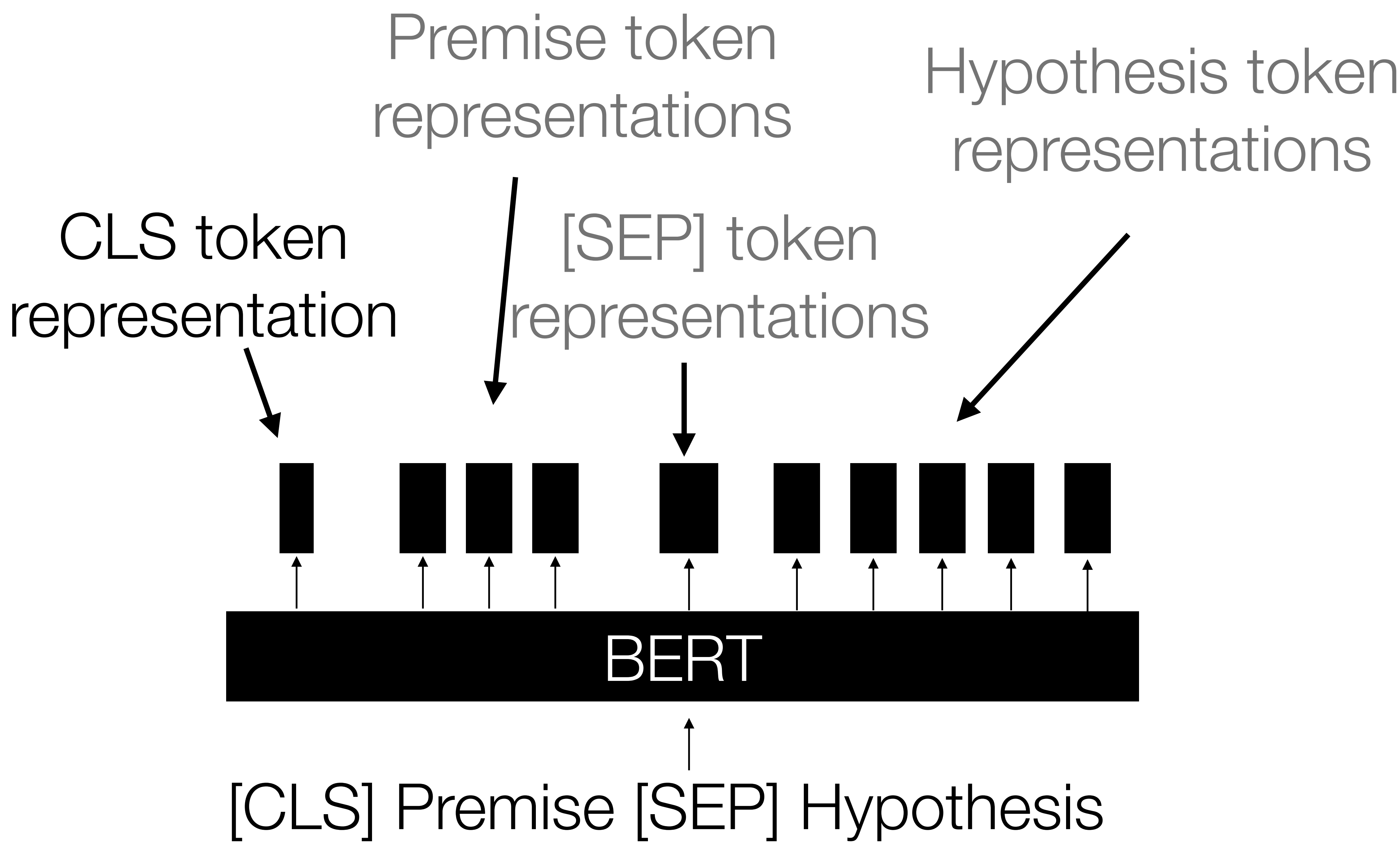
Example: Natural language inference

Example from Stanford Natural Language Inference (SNLI) corpus:

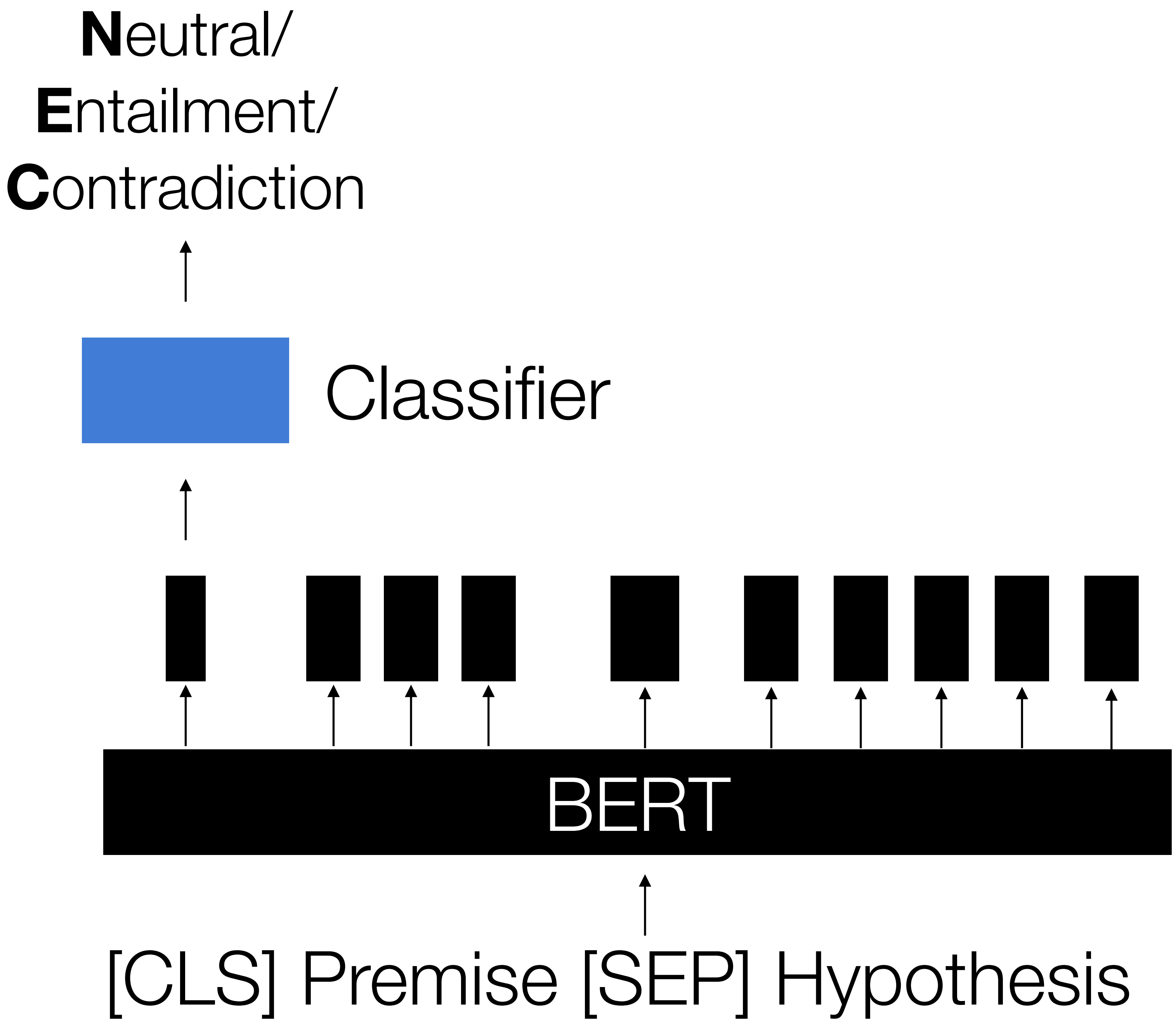
Premise	Three men are sitting near an or-ange building with blue trim.
Entailment	Three males are seated near an orange building with blue trim.
Contradiction	Three women are standing near a yellow building with red trim.
Neutral	Three males are seated near an orange house with blue trim and a blue roof.

Bowman et al. (2015)

NLI with BERT

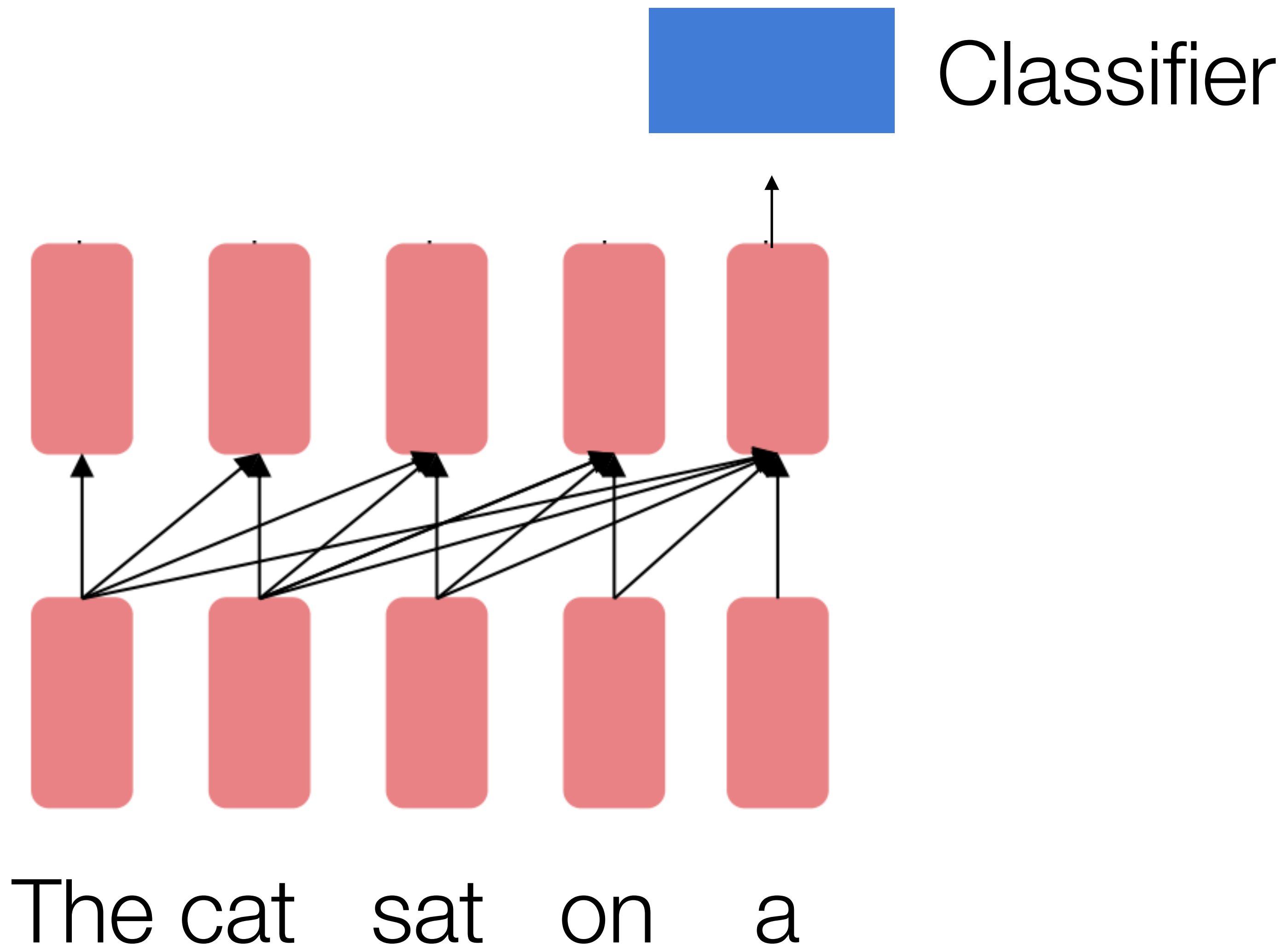


NLI with BERT



Classification with autoregressive LMs

- **Option 1:** Train classifier on top of language model representations



Classification with autoregressive LMs

- **Option 2:** Turn classification into a LM task and continue training language model

Context

Premise: Three men are sitting near an orange building with a blue rim.

Hypothesis: Three males are seated near an orange building.

Label: **Entailment**

Prediction

Autoregressive vs. masked LMs: Summary

- Autoregressive LMs can be used for **generation** or **classification tasks** (or classification through generation)

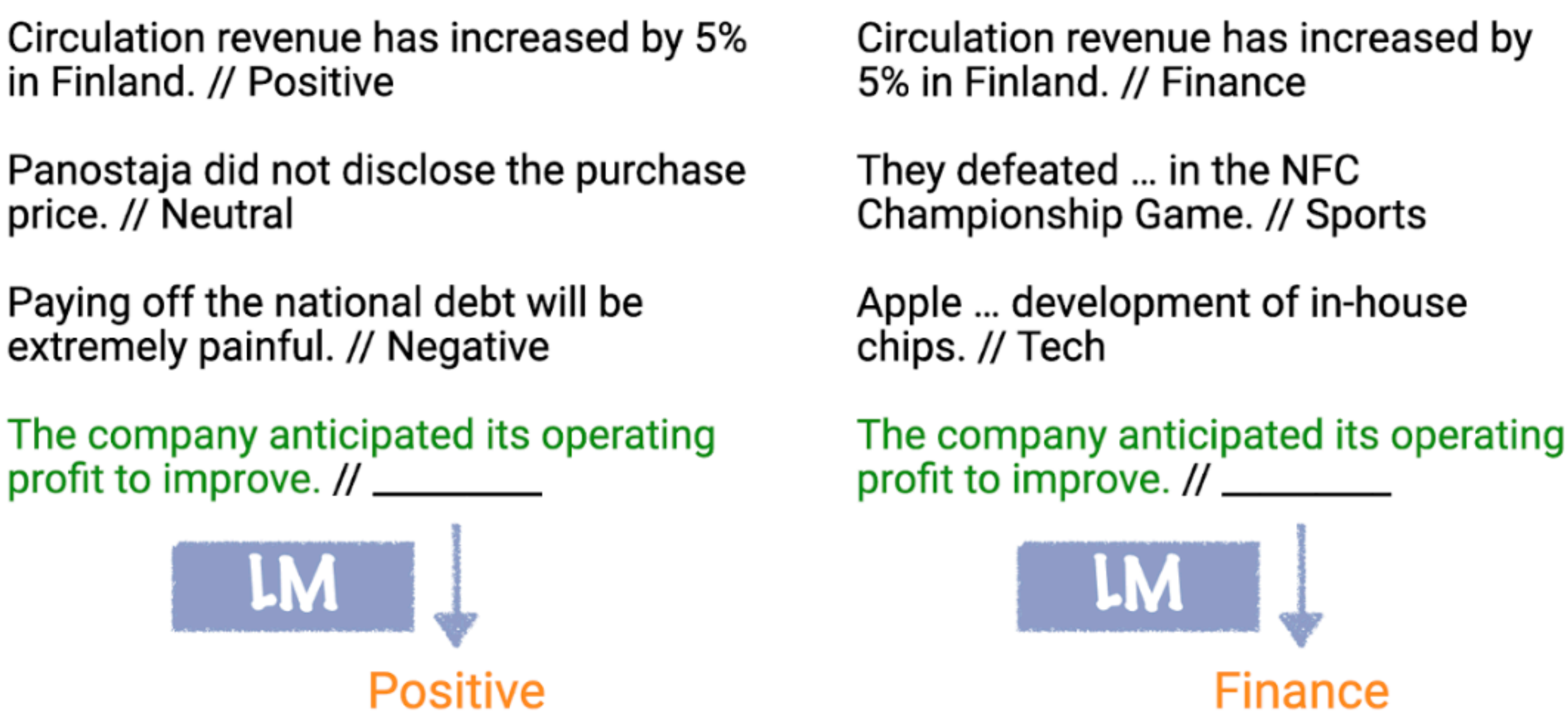
- Masked LMs are primarily used for **sequence classification or sequence labeling tasks**

Prompting and in-context learning

Large pretrained LMs don’t need to be finetuned

- Very large LMs (e.g., GPT-3 w/ 175B parameters) have the ability to do many tasks out of the box or can learn tasks from very few examples that are included in the context
    - “In-context learning”

Classification with in-context learning



http://ai.stanford.edu/blog/understanding-incontext/

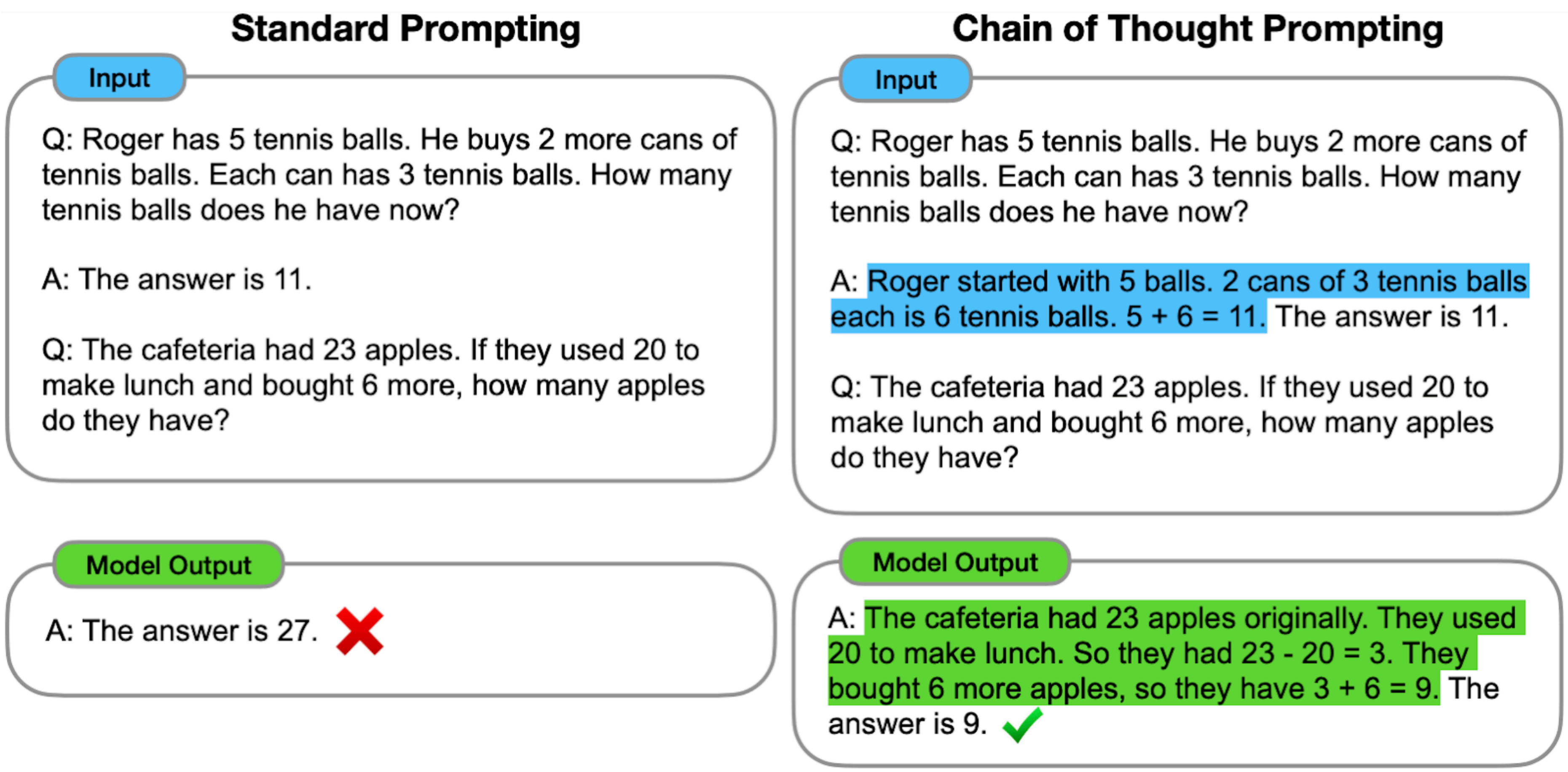
Prompting

- Models like GPT-3 can often also perform tasks reasonably well without any demonstration examples
- The input (“the prompt”) to the model is simply describing what the model should do:

```
Translate the following sentence from English to Spanish.

The cat jumped over the moon.
```

Chain-of-thought prompting

- Multi-step reasoning can be considerably improved by including examples that illustrate intermediate steps:



Nye et al. (2021), Wei et al. (2022)

From GPT-3 to ChatGPT: learning from human feedback

based on slides by Jesse Mu for Stanford CS224N

Language modeling ≠ assisting users

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

COMPLETION GPT-3

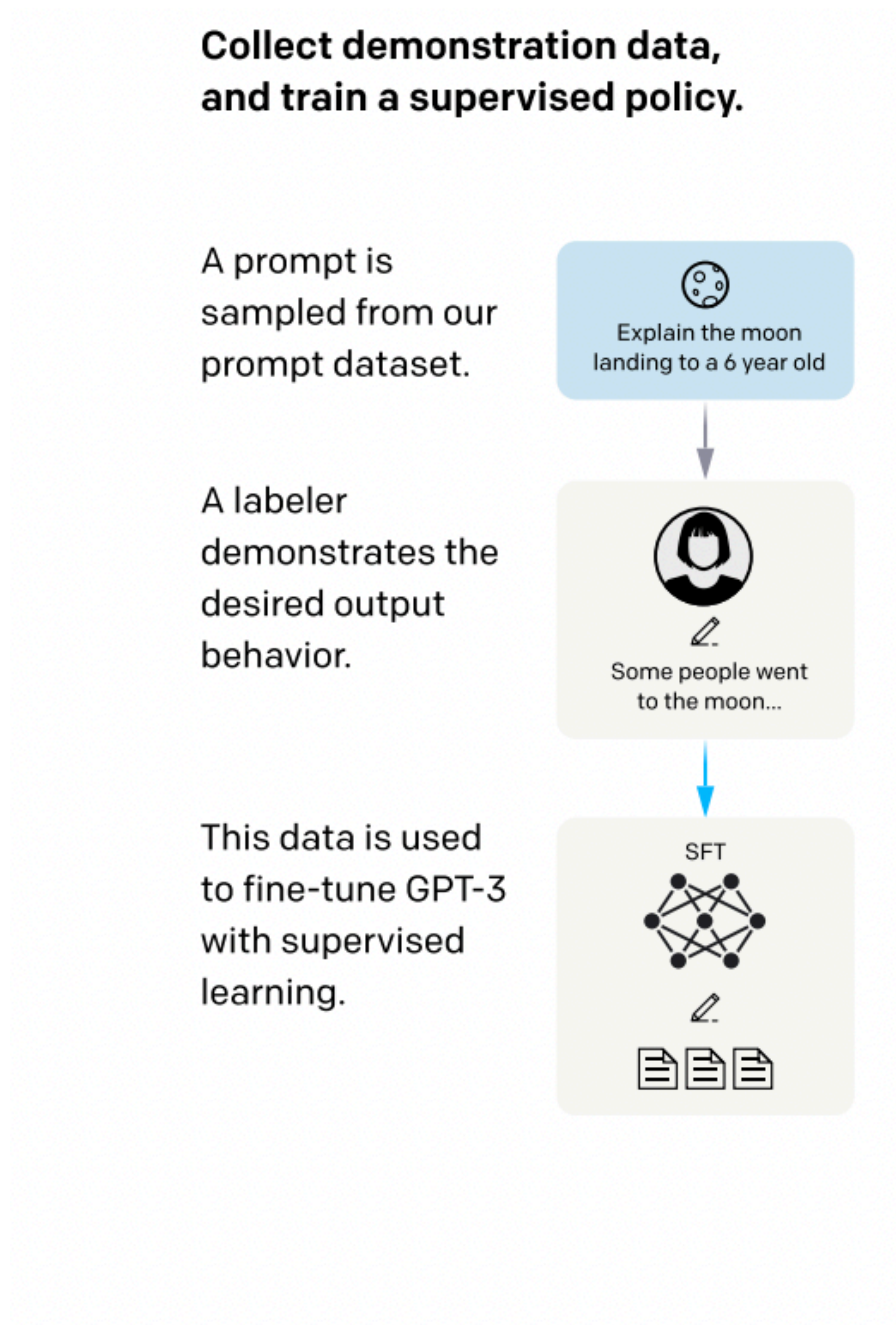
```
Explain the theory of gravity to a 6 year old.

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

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Language modeling ≠ assisting users

PROMPT	*Explain the moon landing to a 6 year old in a few sentences.*
COMPLETION	**Human** A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

Approach 1: Instruction finetuning!

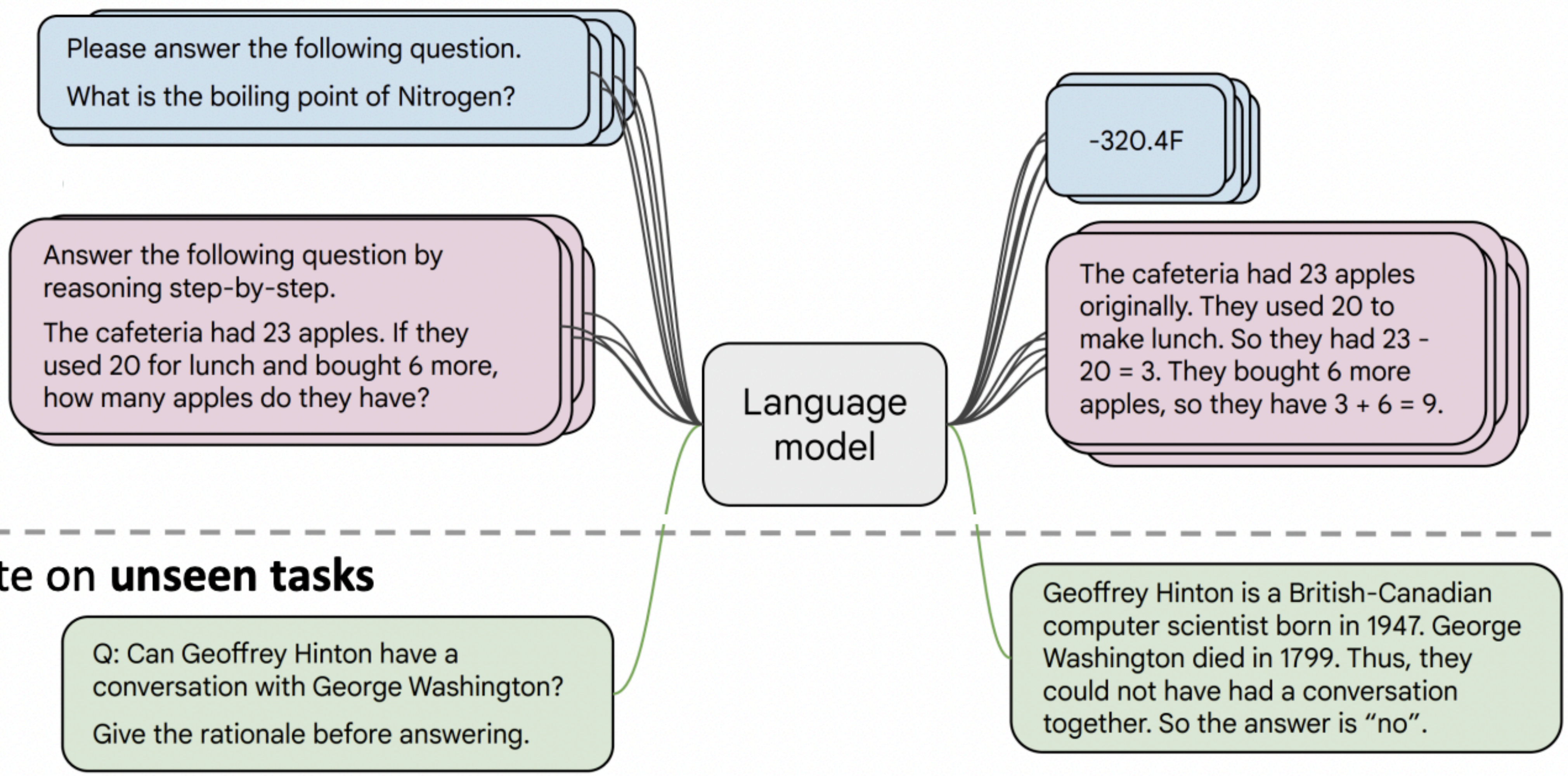
Instruction finetuning



Ouyang (2022)

Instruction finetuning using existing data

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



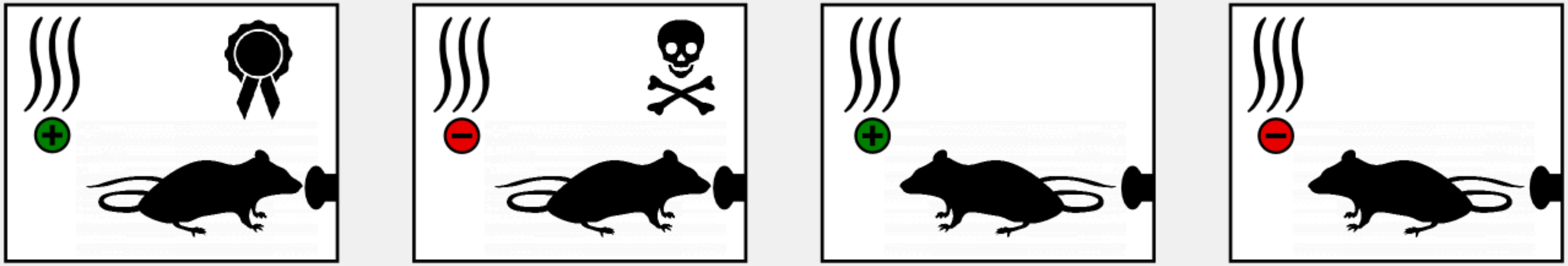
- Evaluate on **unseen tasks**

FLAN-T5; Chung et al., 2022

Problem with instruction finetuning: No unique answers

- Many tasks (e.g. “Write a short story about unicorns”) don’t have a unique expected output
    - Training on next word prediction task is suboptimal
- Idea: Let’s learn from human feedback!

Reinforcement learning



GO + POSITIVE ODOUR (REWARD) | GO + NEGATIVE ODOUR (PUNISHMENT) | NO-GO + POSITIVE ODOUR (NO-PUNISHMENT NO-REWARD) | NO-GO + NEGATIVE ODOUR (NO-PUNISHMENT NO-REWARD)

Optimizing for human preferences

- Let’s say we want to teach a modle to summarize
- We want to “reward" good summaries and “punish” bad summaries

```
SAN FRANCISCO,
California (CNN) --
A magnitude 4.2
earthquake shook the
San Francisco
...
overturn unstable
objects.
```

```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

s_1

$R(s_1) = 8.0$

```
The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.
```

s_2

$R(s_2) = 1.2$

Human preferences

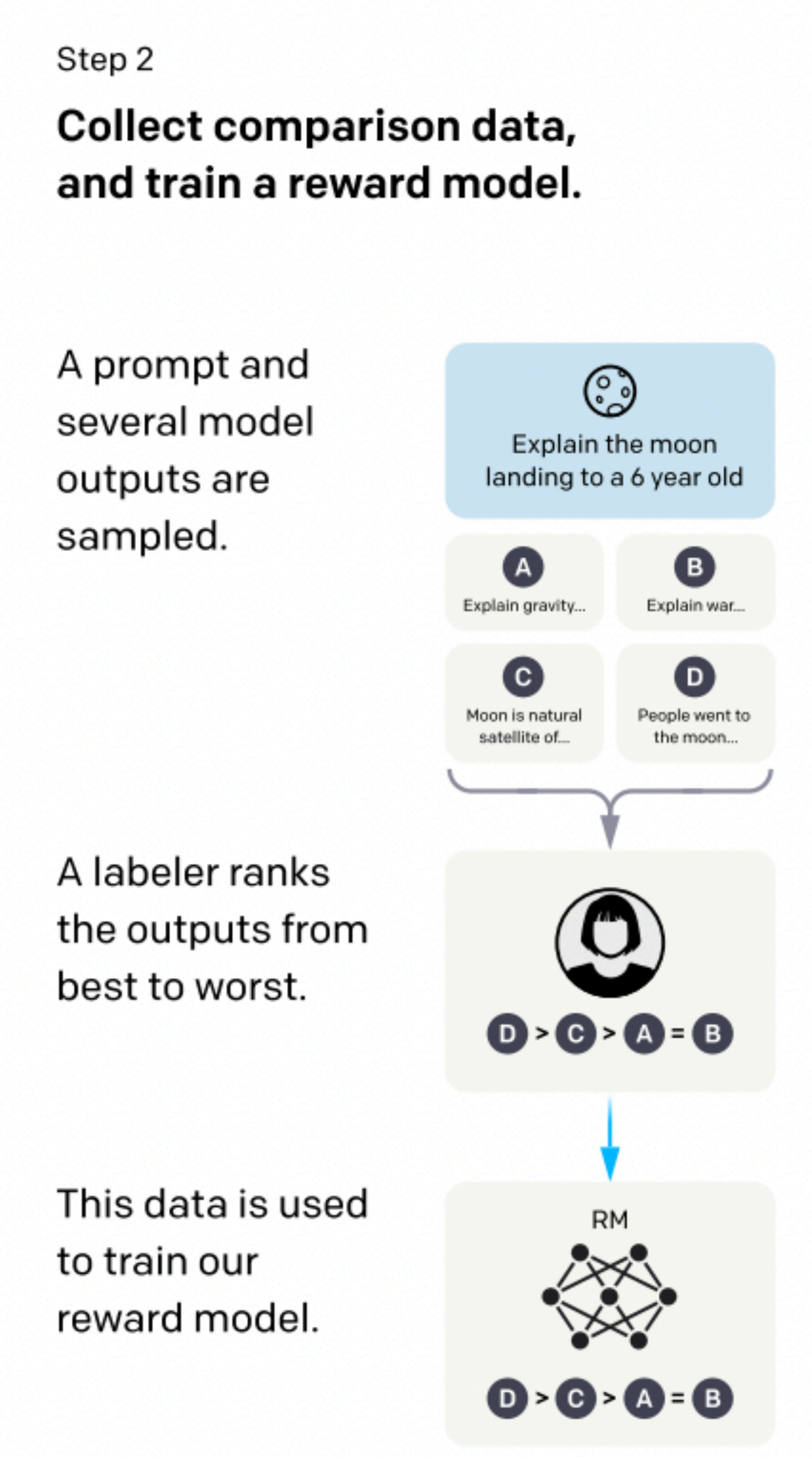
Reward models

- For training a model one needs many many such comparisons between generations

- Human judgements are expensive

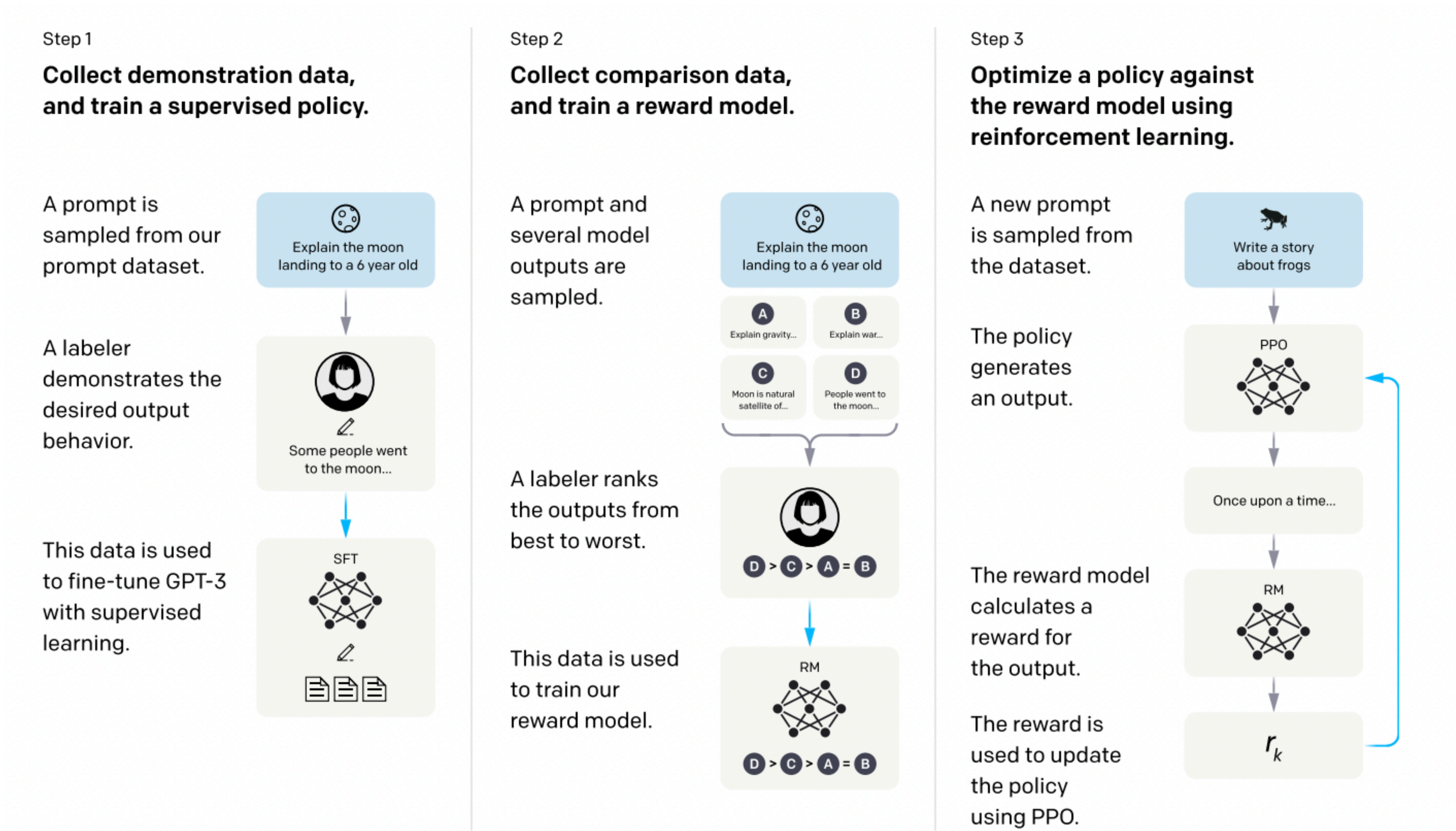
- Idea: Train a reward model that can approximately mimic human preferences

Reward models



Ouyang (2022)

RLHF: Reinforcement learning from Human Feedback



Ouyang (2022)

Another additional ingredient: Code

- Models like ChatGPT and some variants of GPT-3 are trained on human languages and code

- Still not entirely clear why, but pretraining on code and language seems to further improve abilities of LMs (in general, not just on tasks related to code)

Summary

- Pretrained language models can be adapted for downstream tasks in multiple ways:
  - Pretraining and finetuning paradigm
  - In-context learning and prompting
- Chatbots like ChatGPT have two core ingredients
  - Massive pre-trained language model pre-trained on language and code
  - Fine-tuned through instruction tuning and reinforcement learning from human feedback

Next time: What does it mean to understand and methods for evaluating understanding abilities