Natural Language Processing

LLM alignment

Marco Kuhlmann
Department of Computer and Information Science

LINKÖPING UNIVERSITY

The alignment problem

- Language models are trained to generate text that is similar in nature to the text in their training data.
- There is no incentive in language model training to generate text that is helpful, truthful, ethical, etc.
- As a consequence, language models are not necessarily **aligned** with human intents, preferences, or values.

Aligning to follow instructions

Prompt	Explain the moon landing to a 6 year old in a few sentences.
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.
InstructGPT	People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

Aligning language models to follow instructions (OpenAI, 2022)

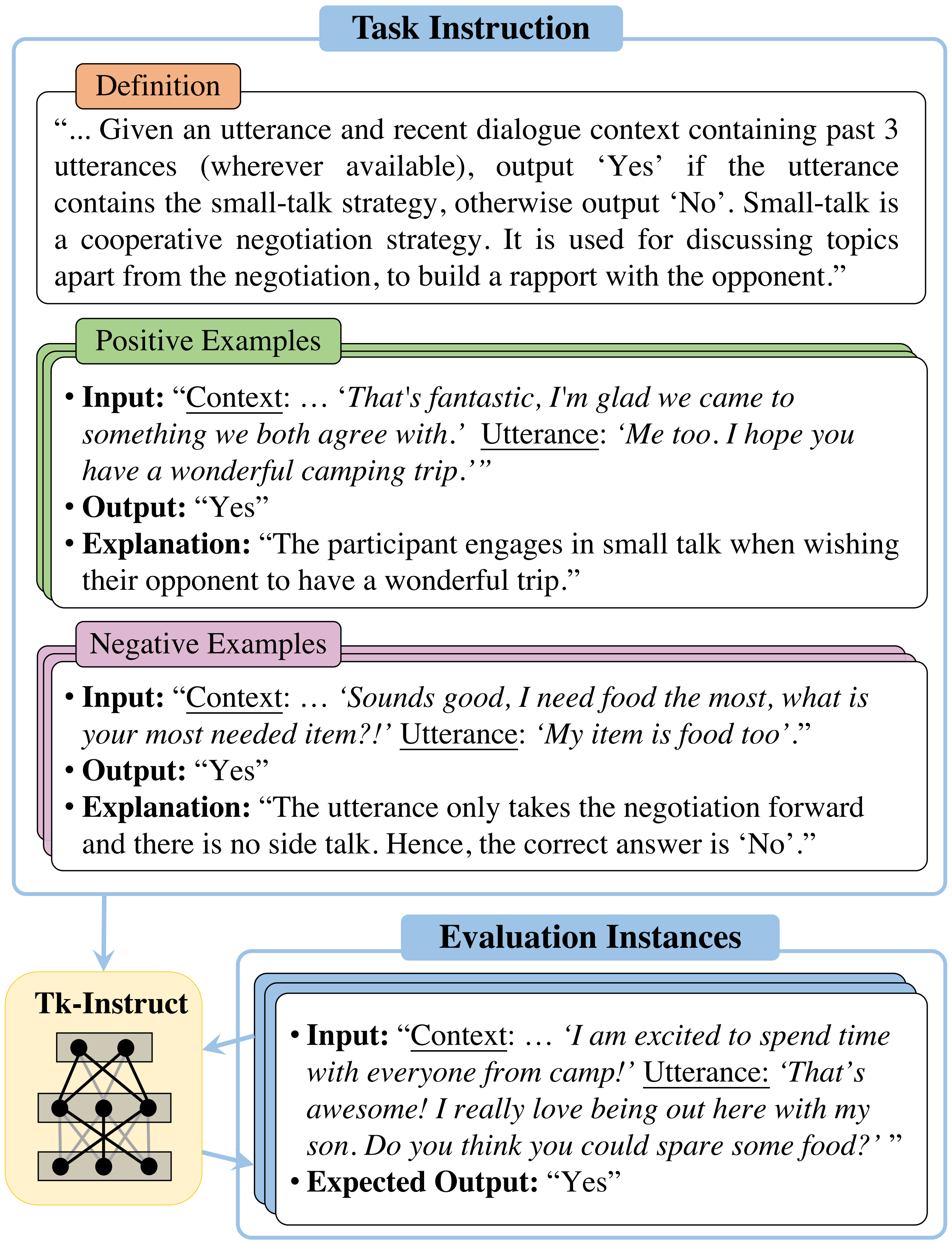
	unsupervised pre-training	instruction fine-tuning	reward modelling	reinforcement learning
data	**raw text from the Internet** billions of words low quality, high quantity	**ideal dialogues** 10k–100k low quantity, high quality		
algorithm	**language modelling** predict the next word	**language modelling** predict the next word		
resources	1000s of GPUs several months of training time GPT, LLaMA	1–100 GPUs several days of training time		

language model → **assistant model**

Instruction finetuning

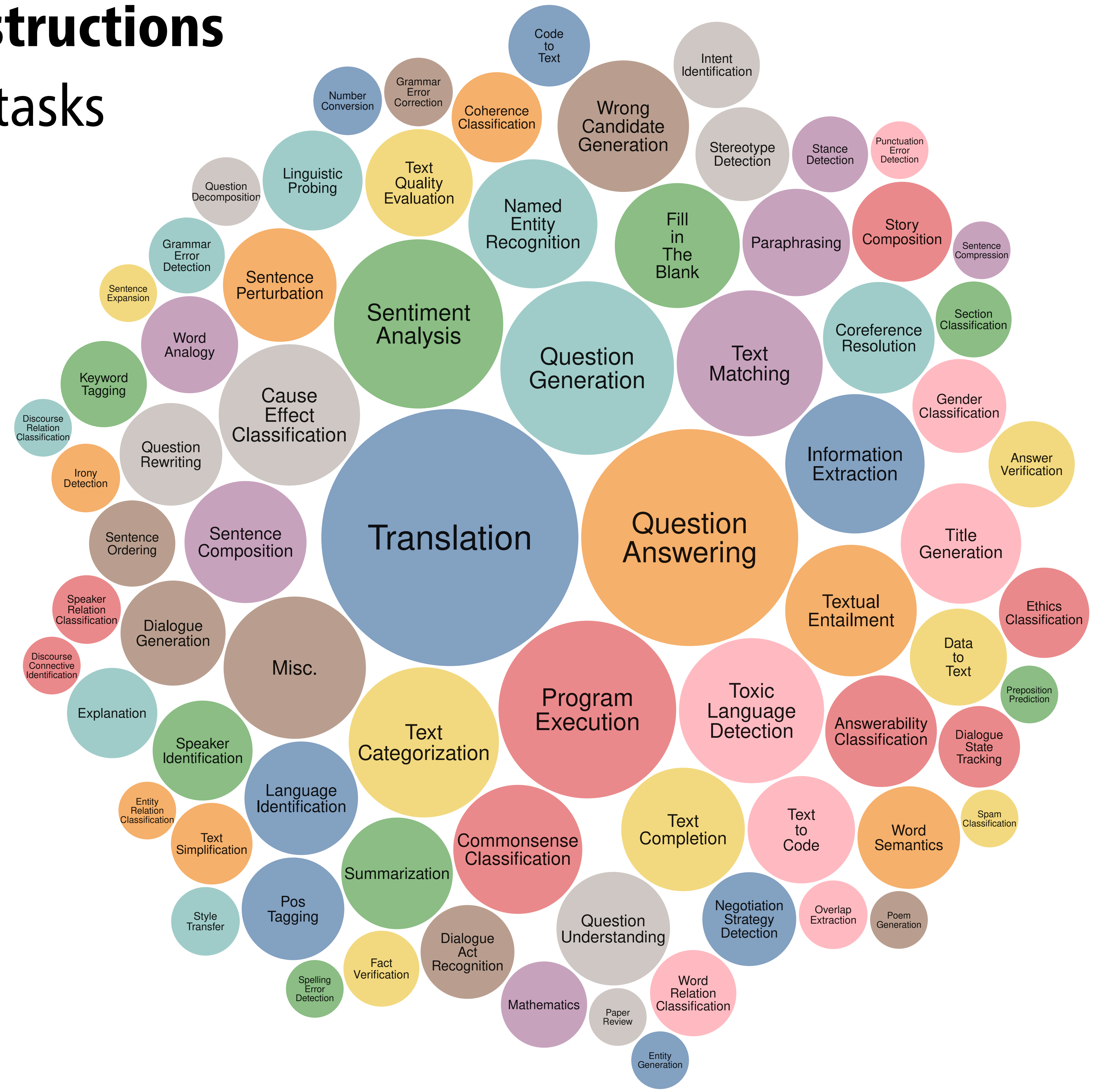
A successful model is expected to use the provided instructions (including task definition and demonstration examples) to output responses to a pool of evaluation instances.

Wang et al., 2022



Super-NaturalInstructions

1,616 diverse NLP tasks



Wang et al. (2022)

Limitations of instruction finetuning

- Collecting ground-truth data for a large number of relevant tasks is expensive and time-intensive.
- There are many tasks that do not have a single correct answer.
- Language modelling as an objective penalises token-level mistakes, but many mistakes are at the conversation level.
- Human preferences are inconsistent.

Credits to Jesse Wu

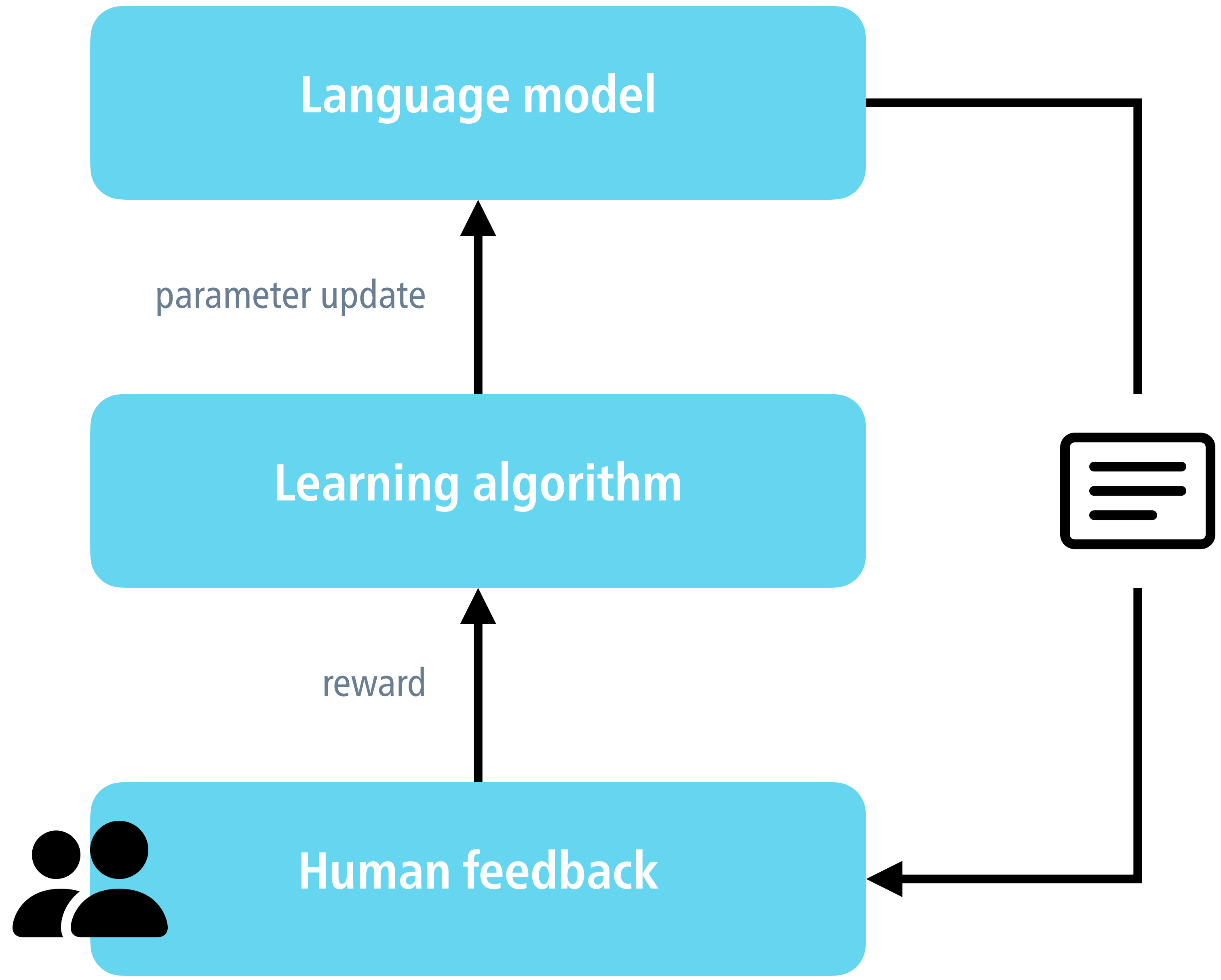
Optimising for human preferences

Prompt: Explain the moon landing to a 6 year old in a few sentences.

Better	Worse
People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.	Explain the theory of gravity to a 6 year old.

Optimising for human preferences

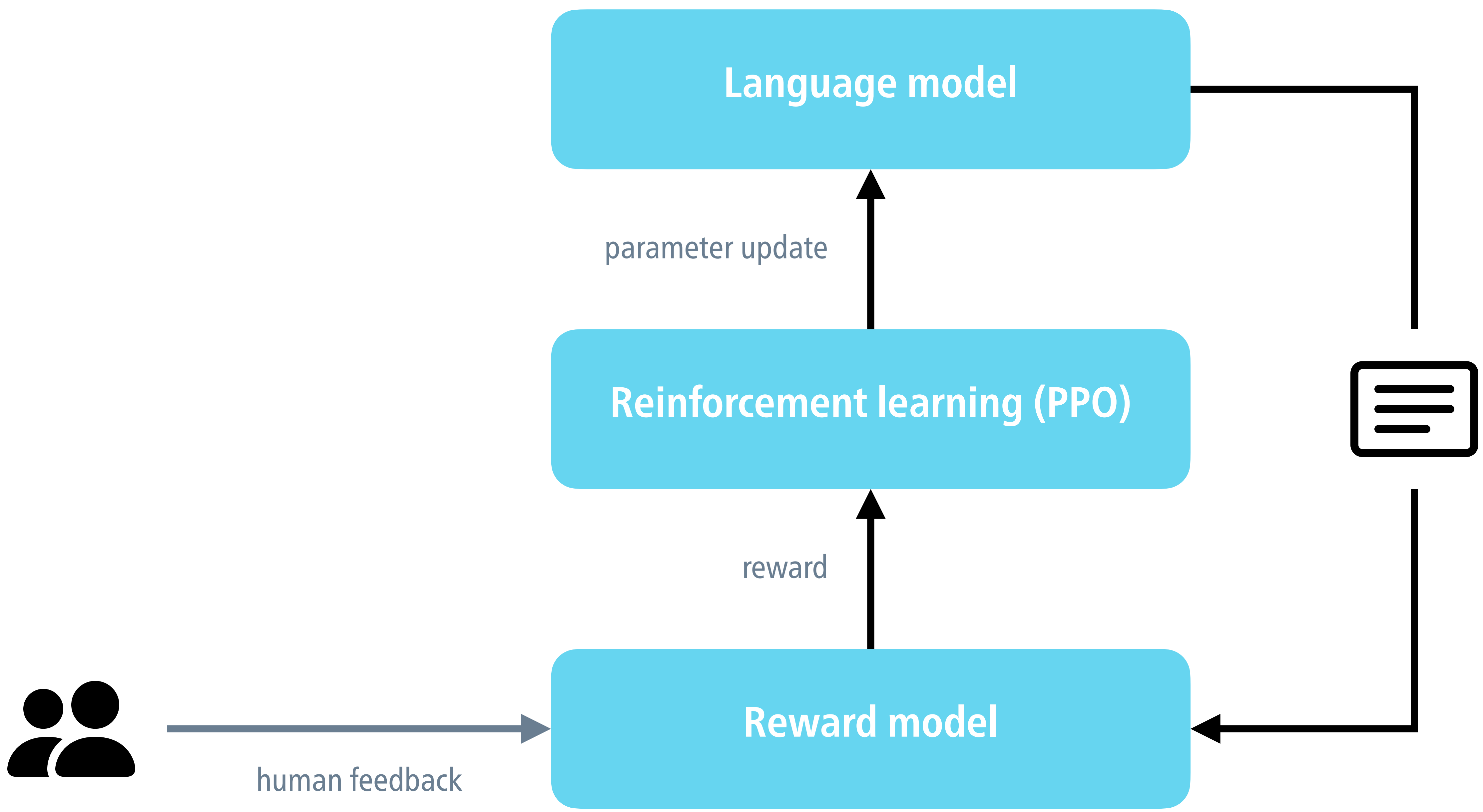
Keeping humans
in the loop
is expensive!



	unsupervised pre-training	instruction fine-tuning	reward modelling	reinforcement learning
data	**raw text from the Internet** billions of words low quality, high quantity	**ideal dialogues** 10k–100k low quantity, high quality	**annotated dialogues** 100k–1M low quantity, high quality	
algorithm	**language modelling** predict the next word	**language modelling** predict the next word	**binary classification** reward consistent with preferences?	
resources	1000s of GPUs several months of training time GPT, LLaMA	1–100 GPUs several days of training time	1–100 GPUs several days of training time	

language model → **assistant model**

Optimising for human preferences



Reward model

- We fine-tune a language model that takes a prompt x and a completion y, and outputs the reward as a scalar.
- For training, we sample m prompt–response pairs and use a cross-entropy loss with the binary human comparisons as labels:

$$\text{loss}(\boldsymbol{\theta}) \;\coloneqq\; -\frac{1}{m}\sum_{i=1}^{m}\log\Big(\sigma\big(\underbrace{R_{\boldsymbol{\theta}}(x_i, y_i^{+})}_{\text{preferred completion}} - \underbrace{R_{\boldsymbol{\theta}}(x_i, y_i^{-})}_{\text{dispreferred completion}}\big)\Big)$$

	unsupervised pre-training	instruction fine-tuning	reward modelling	reinforcement learning
data	**raw text from the Internet** billions of words low quality, high quantity	**ideal dialogues** 10k–100k low quantity, high quality	**annotated dialogues** 100k–1M low quantity, high quality	**generated dialogues** 10k–100k low quantity, high quality
algorithm	**language modelling** predict the next word	**language modelling** predict the next word	**binary classification** reward consistent with preferences?	**reinforcement learning** generate text for maximal reward
resources	1000s of GPUs several months of training time GPT, LLaMA	1–100 GPUs several days of training time	1–100 GPUs several days of training time	1–100 GPUer several days of training time ChatGPT, Claude

language model → **assistant model**

Policy gradient

Williams (1992); Schulman et al. (2017)

- We want to update the parameters of our language model to maximise expected reward.
- To do so, we sample m prompt–response pairs (x_i, y_i), compute rewards according to our reward model, and do gradient ascent:

$$\boldsymbol{\theta}_{t+1} \ := \ \boldsymbol{\theta}_t + \alpha \frac{1}{m} \sum_{i=1}^{m} R(x_i, y_i) \nabla_{\boldsymbol{\theta}_t} \log p_{\boldsymbol{\theta}_t}(y_i \mid x_i)$$

reward is positive – take gradient steps to maximise probability

reward is negative – take gradient steps to minimise probability

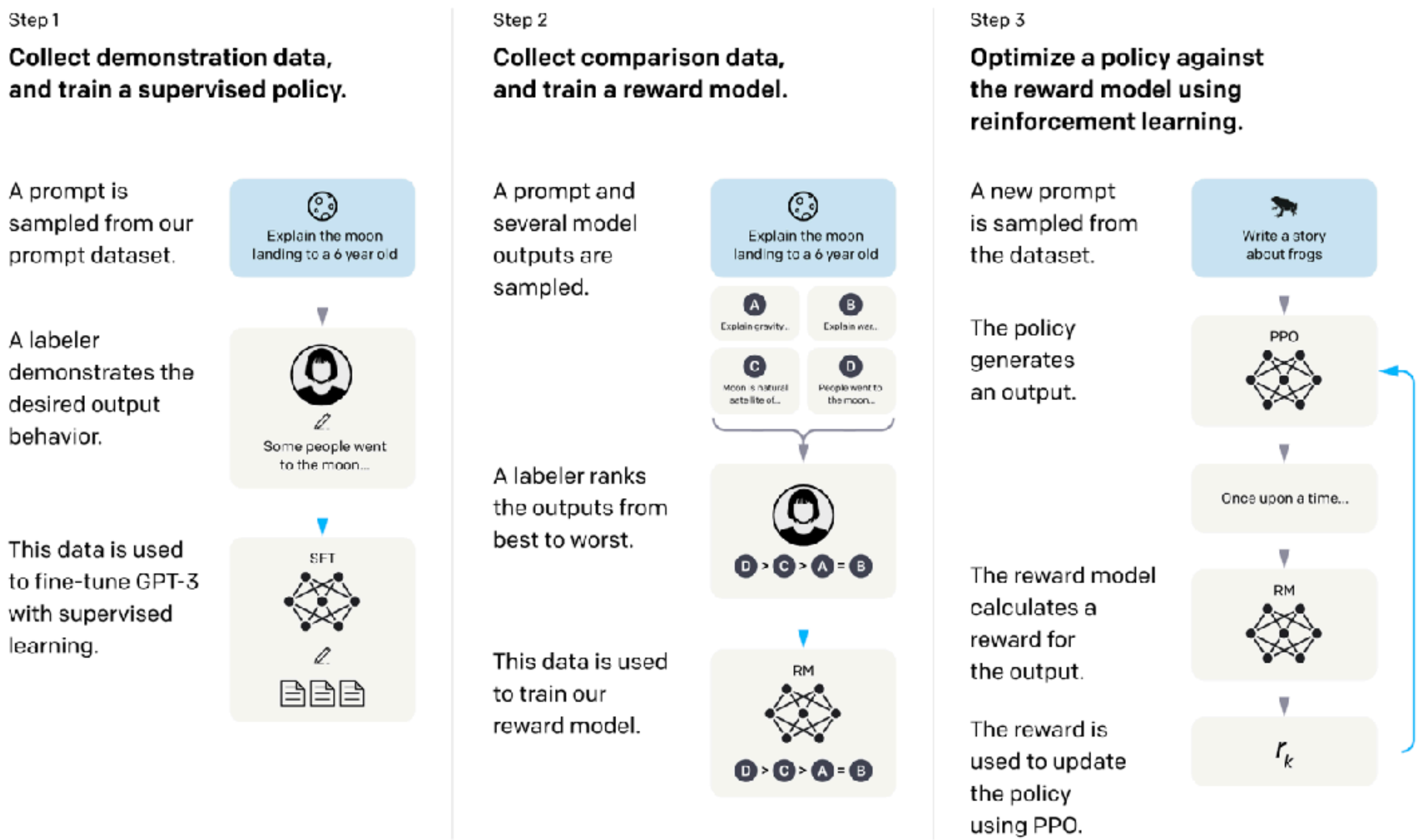


Figure 2 from Ouyang et al. (2022)

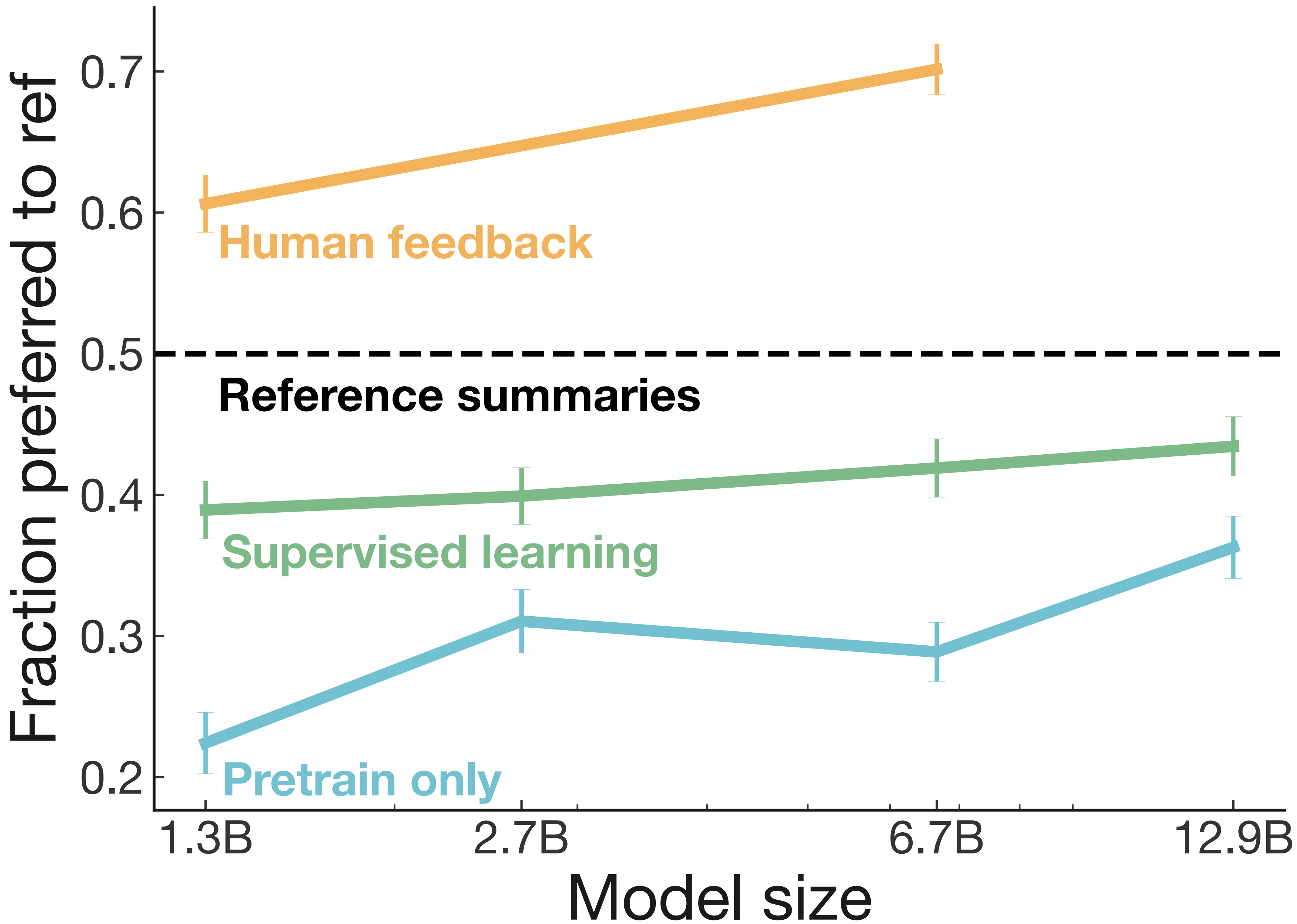
Putting it all together

Ouyang et al. (2022); Schulman et al. (2017)

- Starting from the finetuned language model p^{FT}, we obtain updated language models p^{RL} using policy gradient methods.
- To penalise the updated models for diverging too far from the finetuned model, we use a modified reward function:

$$\underbrace{R'(x, y)}_{\text{sample}} = \underbrace{R(x, y)}_{\text{reward model}} - \underbrace{\beta \log\left[p_{\boldsymbol{\theta}}^{\mathrm{RL}}(y \mid x) / p^{\mathrm{FT}}(y \mid x)\right]}_{\text{penalty based on KL divergence}}$$

Effectiveness of human feedback



Stiennon (2020)