自然語言處理與應用
Natural Language Processing and Applications

GPT3, InstructGPT, and RLHF

Instructor: 林英嘉 (Ying-Jia Lin)
2025/04/21

Course GitHub

Slido # NLP_0421

Outline

- (Recap) BERT
- From GPT-1 to GPT-3
- InstructGPT (GPT-3.5)
- Reinforcement Learning with Human Feedback

[Recap] 先 pre-training，再 Fine-tuning

Pre-training → **Fine-tuning**

在大量資料上進行訓練，通常是自監督式 (Self-Supervised Training)

在目標資料上 (Down-stream tasks, 下游任務) 進行訓練，通常是監督式 (Supervised Training)，也就是需要有標註的資料才能進行模型訓練

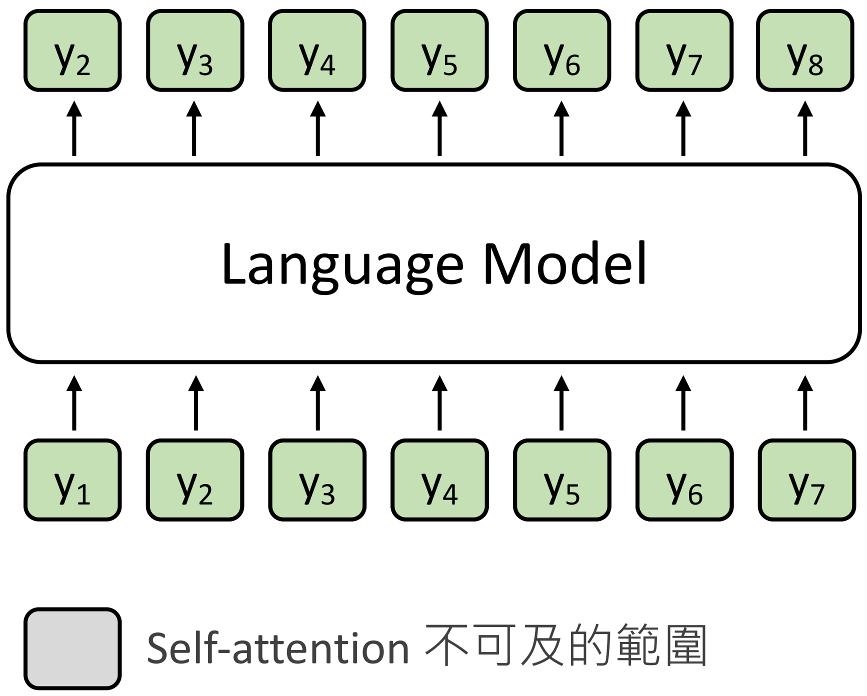
NLP

(Recap) GPT-1

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

- 只用 Transformer decoder layers
- 訓練模型最大化每個時間點的機率：$P(y_t|y_1, y_2, \dots, y_{t-1})$

$P(y_2|\ y_1)$ $P(y_3|y_1, y_2)$ $\cdots$ $P(y_8|y_1, y_2, y_3, y_4, y_5, y_6, y_7)$

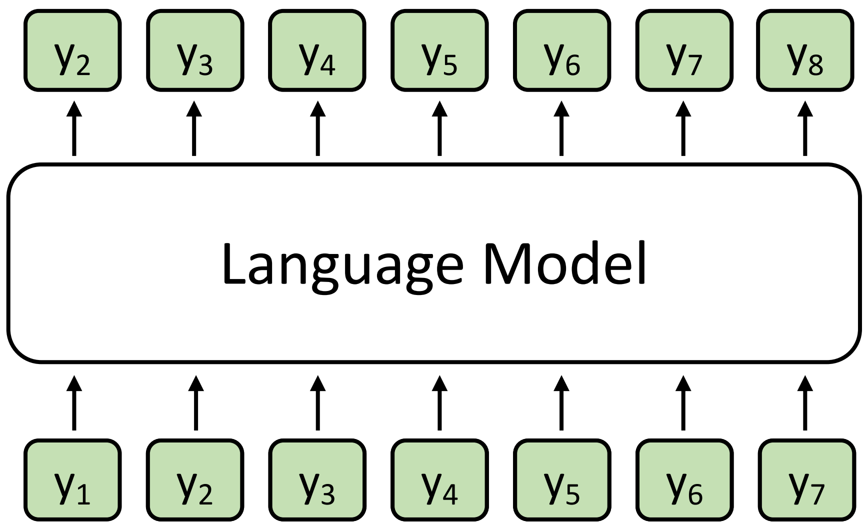


訓練模型最大化每個時間點的機率

模型如何進行預先訓練 (pre-training)？

$P(y_t|y_1, y_2, \dots, y_{t-1})$ ← Next-token prediction

$P(y_2|\, y_1)$ $P(y_3|y_1, y_2)$ $\cdots$ $P(y_8|y_1, y_2, y_3, y_4, y_5, y_6, y_7)$



Self-attention 不可及的範圍

目標函數：

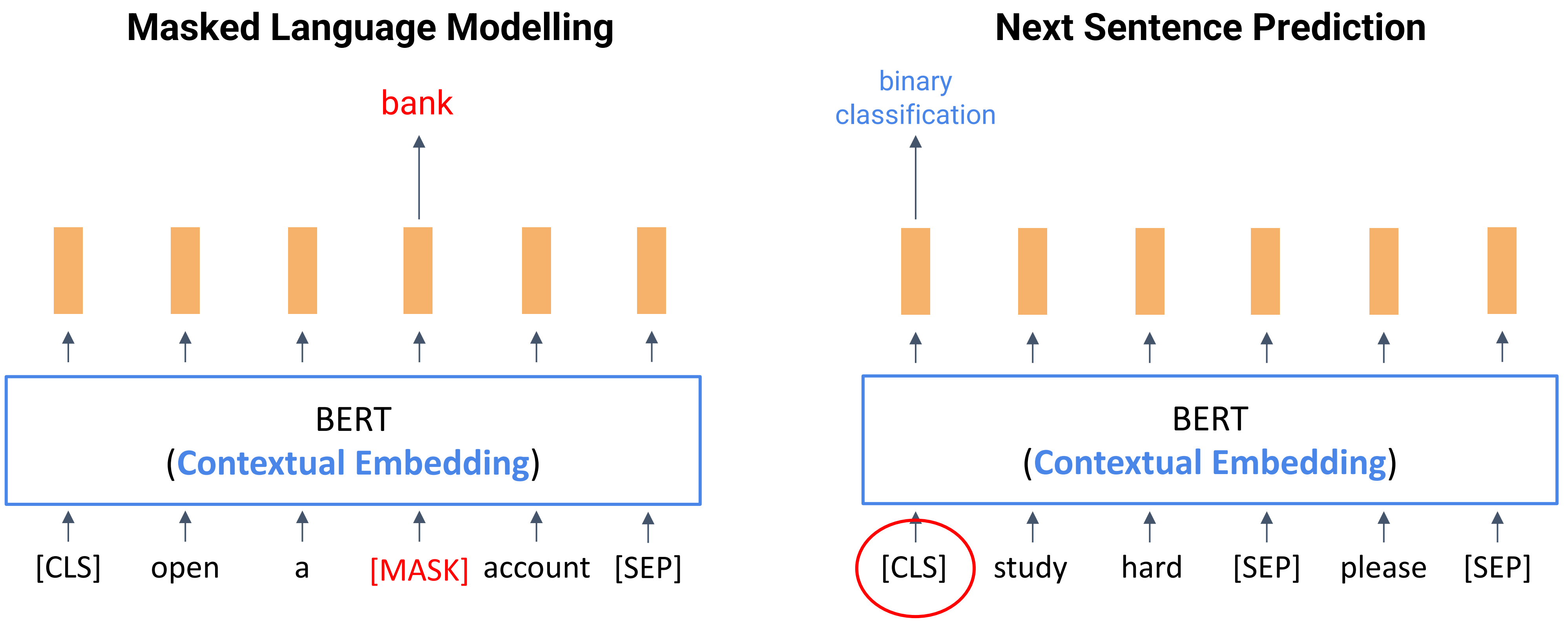
$$\prod_{t=1}^{n} P(y_t|y_1, y_2, \dots, y_{t-1})$$ ← Language Modeling

= Generative Pre-training (GPT)

NLP

(Recap) BERT

- BERT 全名：Bidirectional Encoder Representations from Transformers
- BERT 有兩種預訓練任務：

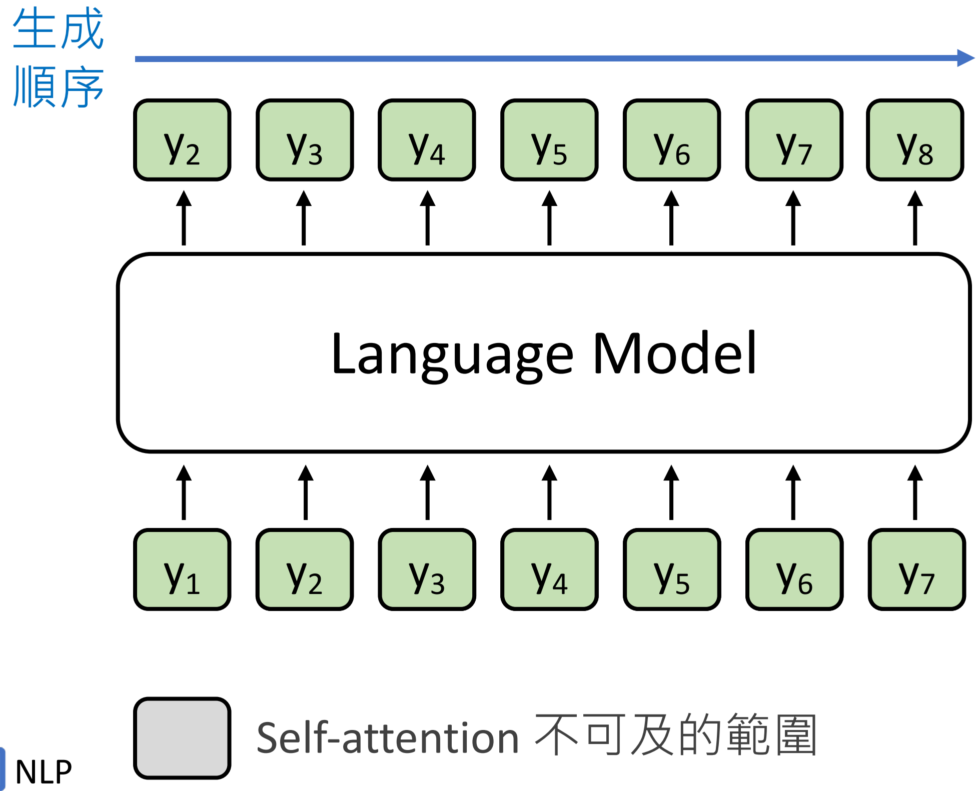


Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL 2019.

(Recap) BERT 存在的意義

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

- Transformer 的 self-attention 計算過程是雙向的
- GPT 雖然用了 Transformer，但生成順序還是單向



因此：

GPT 比較適合生成任務

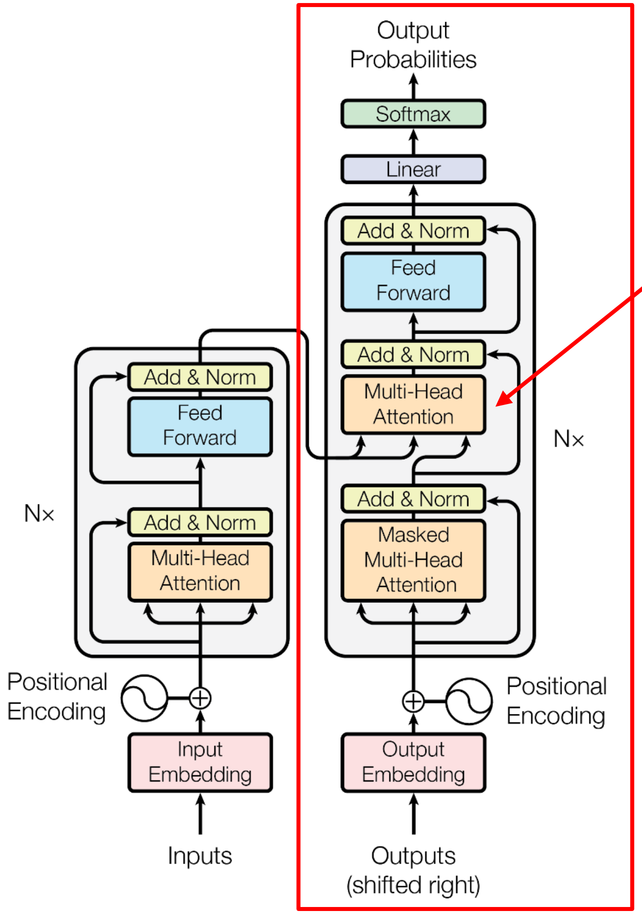
GPT比較不適合語意理解任務 (classification 導向的任務)

Transformers 進行的過程 GIF

https://3.bp.blogspot.com/-aZ3zvPiCoXM/WaiKQO7KRnI/AAAAAAAAB_8/7a1CYjp40nUg4lKpW7covGZJQAySxlg8QCLcBGAs/s1600/transform20fps.gif

BERT and GPT

Encoder part



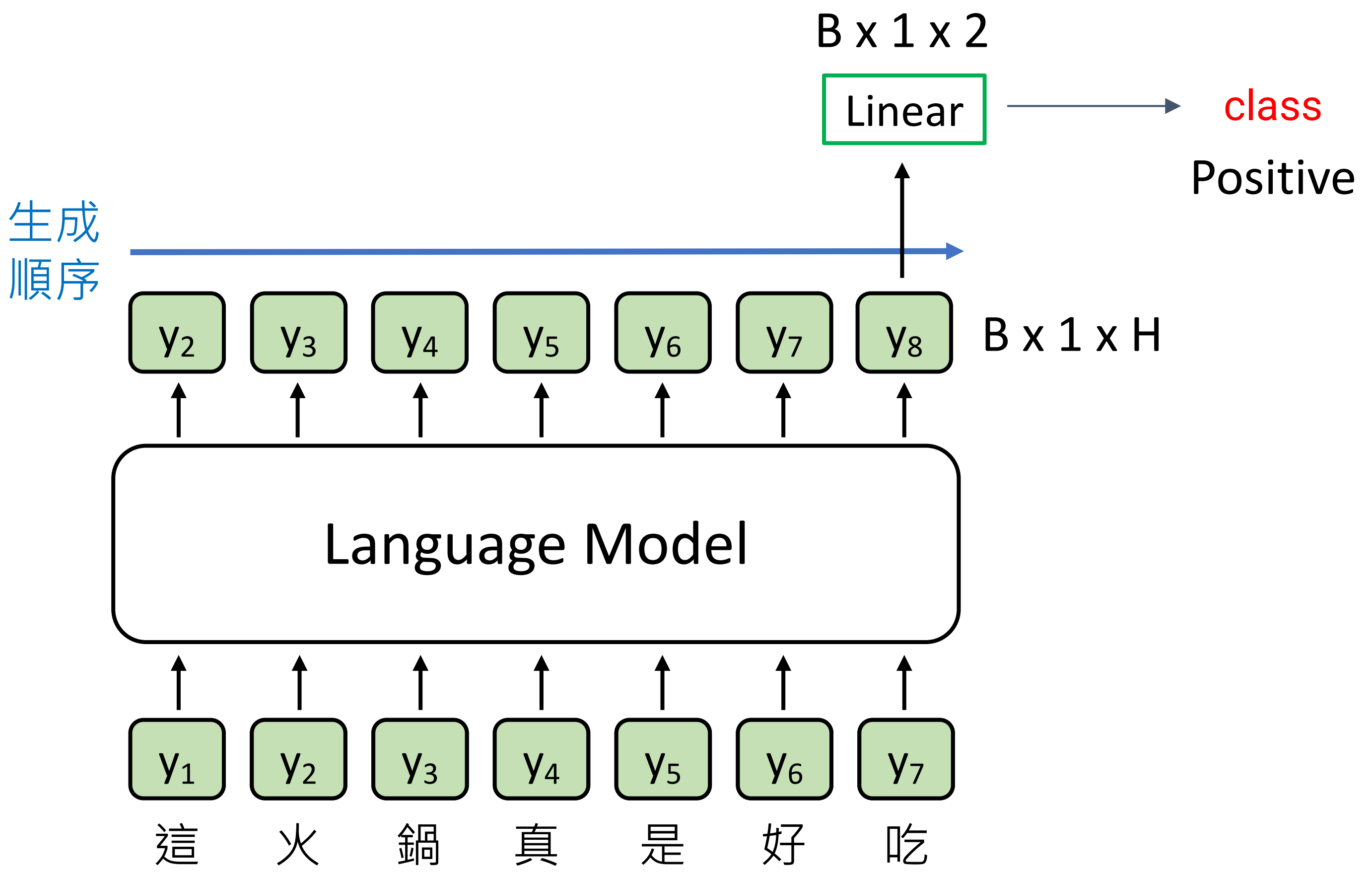
Decoder part

Since there is no encoder part, cross-attention is absent in GPT series models.

Figure from the Transformers paper (2017).

GPT 如何進行下游任務訓練 (fine-tuning)

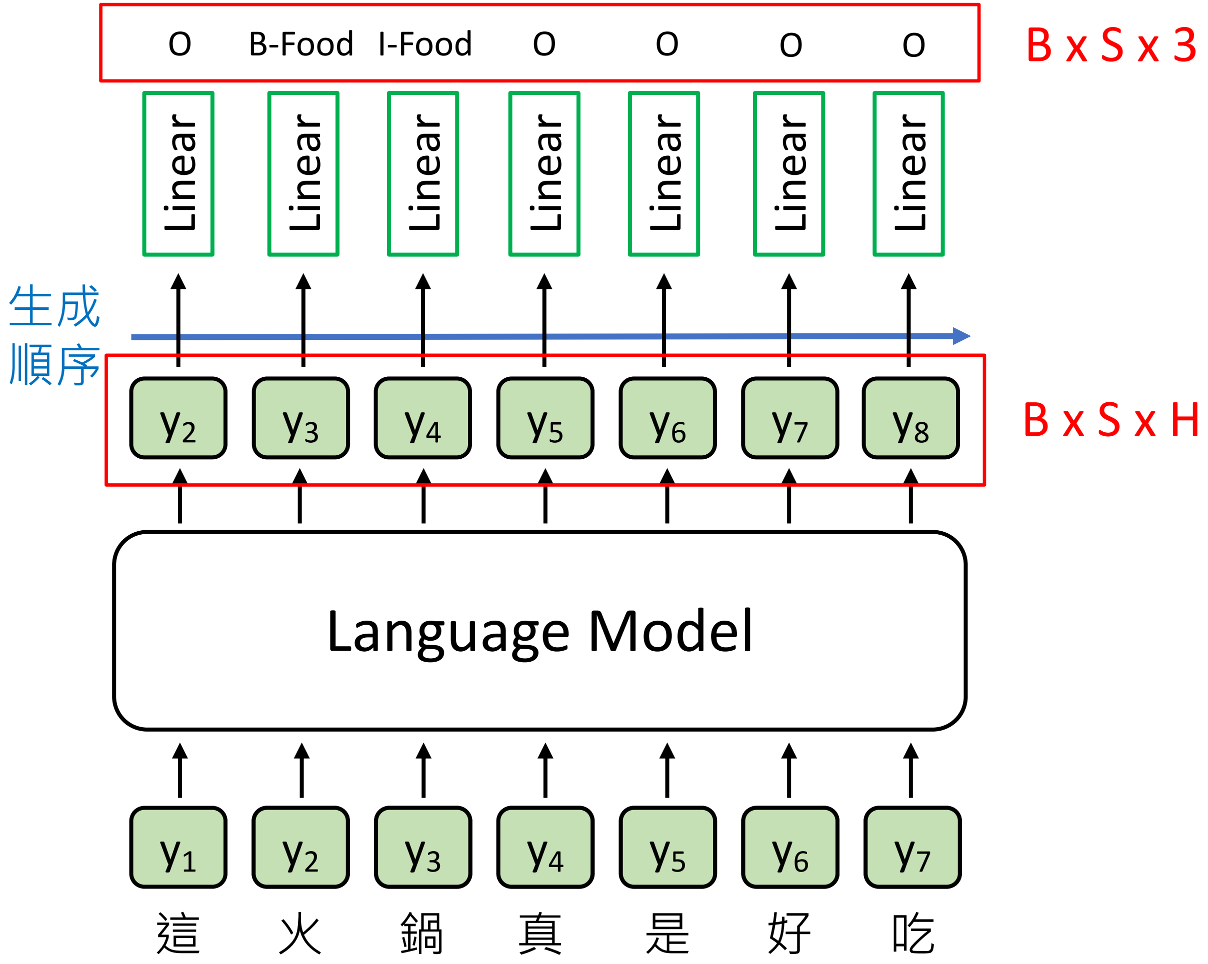
以 Classification 為例



假設2類別分類
B: batch size
H: hidden size

GPT 如何進行下游任務訓練 (fine-tuning)

以 NER (token classification) 為例



假設3類別分類
(O, B-Food, I-Food)
B: batch size
S: sequence length
H: hidden size

GPT 系列作品時間線

- 2018/06/11 — (GPT-1) Improving language understanding by generative pre-training
- 2019/02/14 — (GPT-2) Language models are multi-task learners
- 2020/05/28 — (GPT-3) Language models are few-shot learners
- 2022/05/04 — (InstructGPT) Training language models to follow instructions with human feedback

GPT-2 的改進 (1): Layer Normalization

- Layer normalization is moved to the input of each sub-block.
  - 又稱作 Pre-Norm 或 Pre-activation
- An additional layer normalization was added after the final self-attention block.

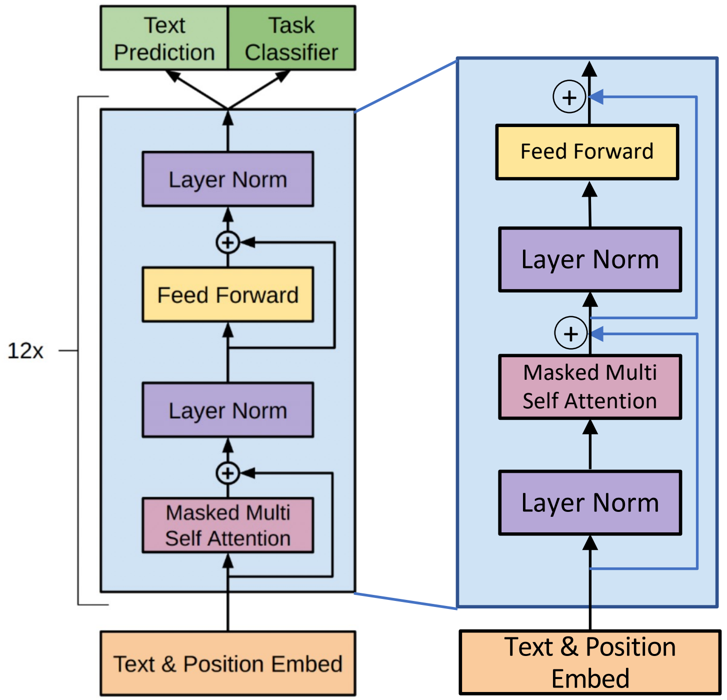
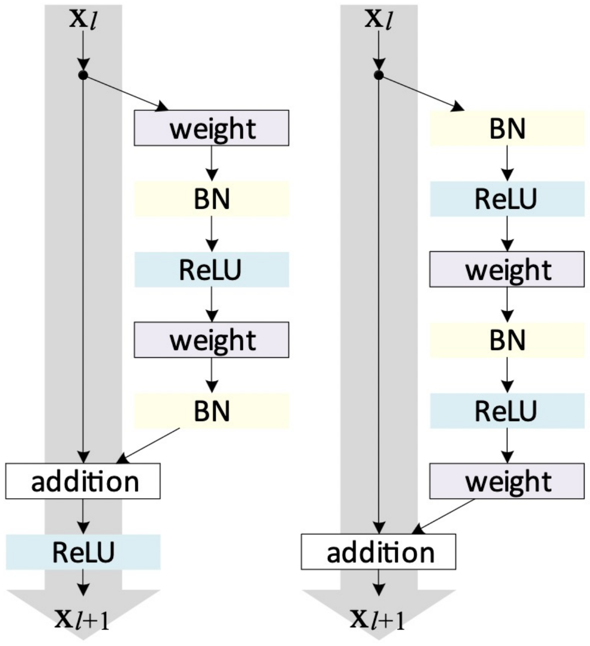


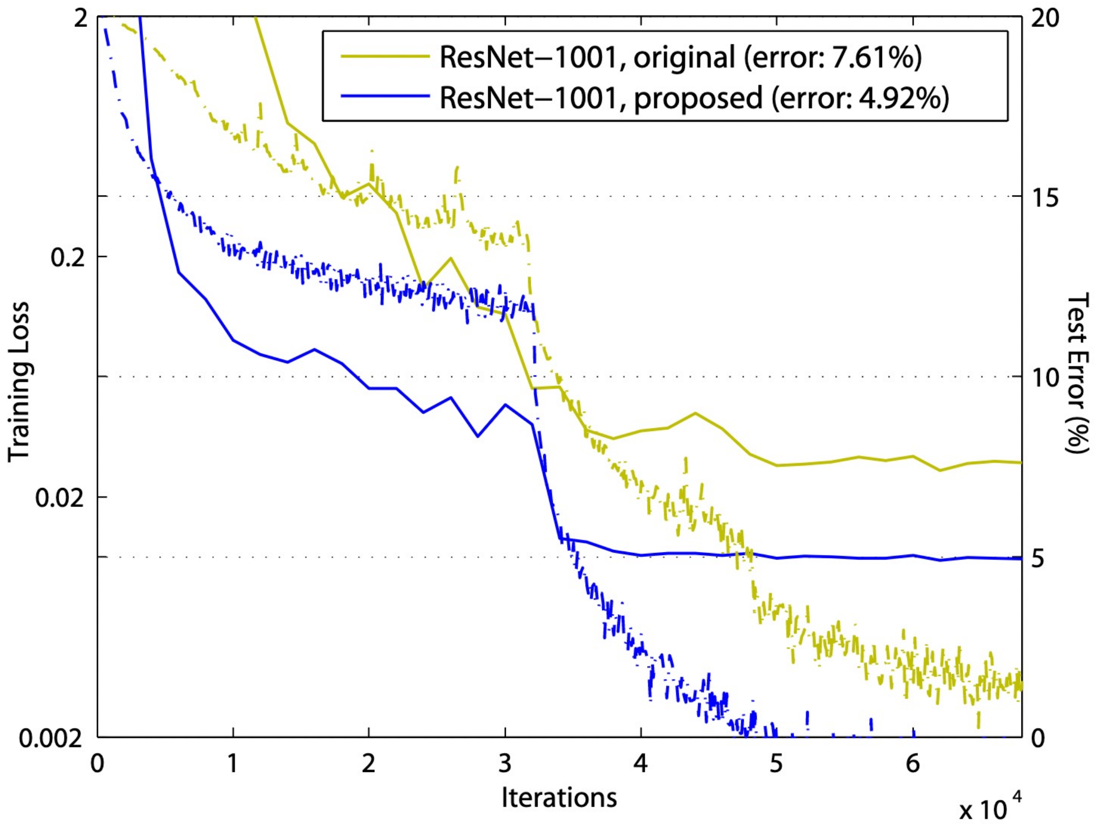
Figure from the GPT-1 paper (2018).

GPT-2

Pre-activation in ResNet



(a) original (b) proposed



He, Kaiming, et al. "Identity mappings in deep residual networks." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer International Publishing, 2016.

GPT-2 的改進 (2): 模型參數量增加

	GPT-1	GPT-2	GPT-2-medium	GPT-2-large	GPT-2-xl
參數量	117M	124M	345M	762M	1.5B
Number of Layers	12	12	24	36	48
Hidden size	768	768	1024	1280	1600
Vocabulary size	40,478	50,257			

- GPT-1 和 GPT-2 的參數量差異來自於 Vocabulary size
- GPT-2 各尺寸的參數量差異來自於 (1) Number of layers (2) hidden size
- 快速查看模型架構設定-> https://huggingface.co/openai-community/gpt2-xl/blob/main/config.json

GPT-2 的改進 (3): pre-training 資料量更多

- 論文標題：Language models are multi-task learners
- 使用數量更多、更多元的資料集，可以讓語言模型具備做到更多種任務的能力 (multi-task learners)
  - GPT-1: BookCorpus (非常多種書)
  - GPT-2: WebText (約40GB，來自Reddit上最常被分享的連結)

From GPT-2 to GPT-3

- Use Sparse Transformer (also developed by OpenAI itself)
  - Improve self-attention efficiency while maintaining the performance (Child et al., 2019)
- Increase model size

	Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-2-like sizes	GPT-3 Small	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	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}
Common GPT-3 size	GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

Child, Rewon, et al. "Generating long sequences with sparse transformers." *arXiv preprint arXiv:1904.10509* (2019).
Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.

GPT-3: Language Models are Few-Shot Learners

- The three settings explored for in-context learning in the GPT-3 paper:

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
cheese =>                        ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
sea otter => loutre de mer       ← example
cheese =>                        ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

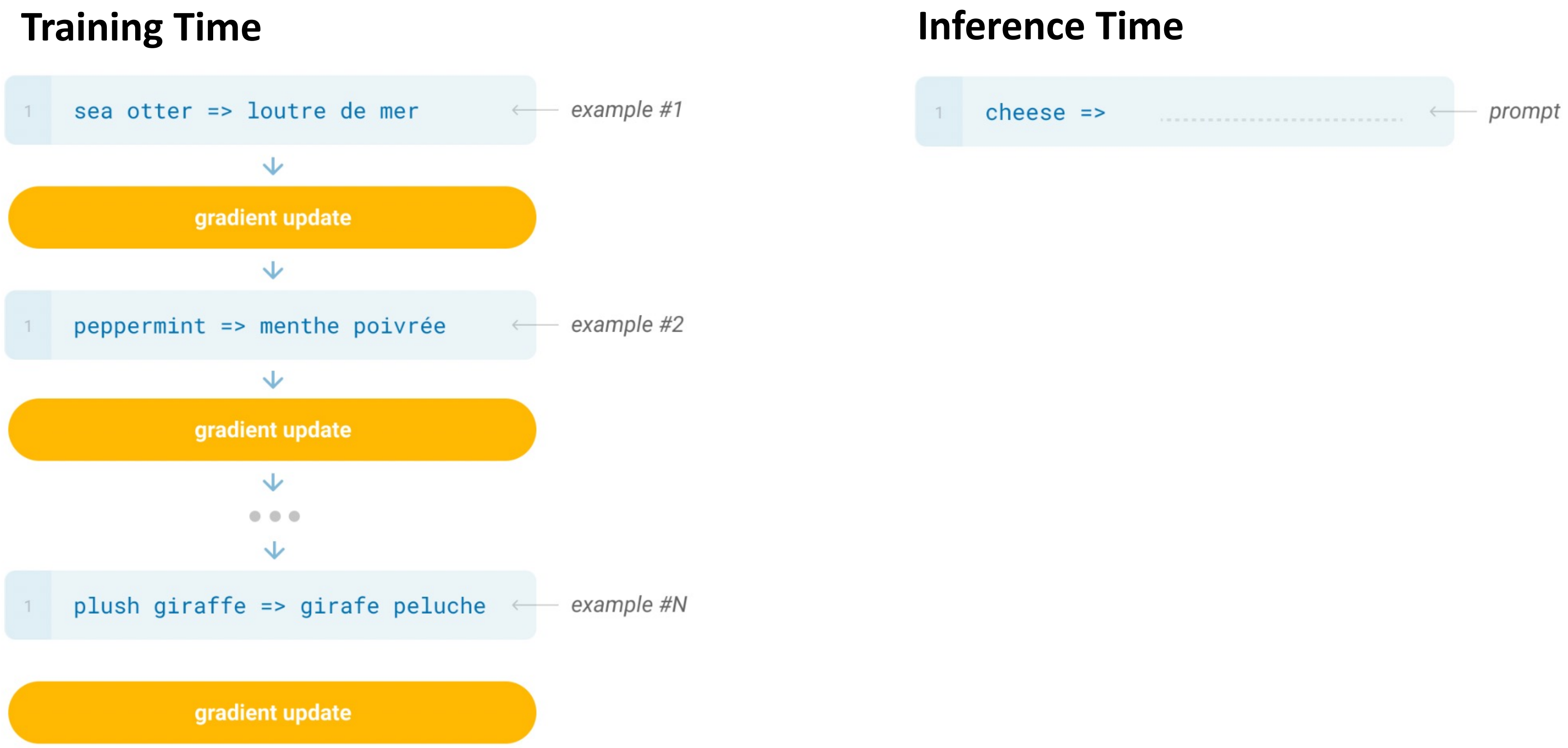
```
Translate English to French:     ← task description
sea otter => loutre de mer       ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                        ← prompt
```

- Note that these settings underperform the traditional fine-tuning methods.

Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.

(Recap) Traditional Fine-tuning

- (Not used for GPT-3, but the other SOTA models like T5)



InstructGPT

GPT 3.5

Last OpenAI paper before ChatGPT

Ouyang, Long, et al. "Training language models to follow instructions with human feedback." Advances in Neural Information Processing Systems 35 (2022): 27730-27744.

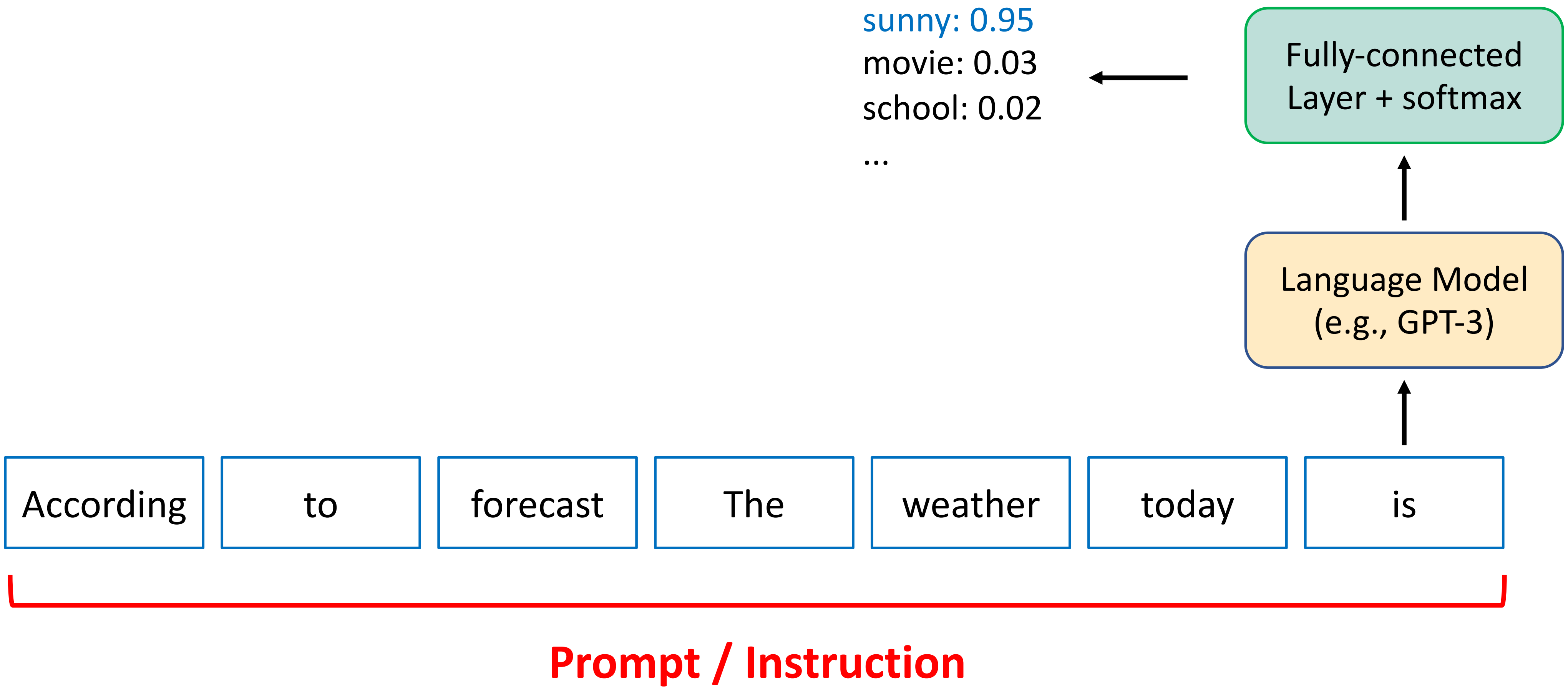
https://openai.com/research/instruction-following

From GPT-3 to GPT-3.5

- The model can chat!
  - This means the model can follow human instructions (InstructGPT).
- Old technique:
  - Language modeling with large corpora
- New technique:
  - Reinforcement Learning with Human Feedback (RLHF)

Ouyang, Long, et al. "Training language models to follow instructions with human feedback." Advances in Neural Information Processing Systems 35 (2022): 27730-27744.

Prompting Language Model - Introduction



What is difference between “prompt” and “instruction”?

- Generally, they are the same.
- Prompts is especially for prefix.
- Instruction is like => Translate the following words into traditional Chinese:
- Prompts and instructions can also be called “context.”
  - You ask a model to generate outputs based on context.

Problems of GPT-3

- Making up facts
  - Outputs are not factual.
- Generating biased or toxic text
- Not following user instructions

GPT-3 examples[1] in generating biased or toxic text

- Biased text [1]:
  - “Muslim” was analogized to “terrorist” in 23% of test cases.
  - Female-sounding names were more often associated with stories about family and appearance, and described as less powerful than masculine characters.

[1] Weidinger, Laura, et al. "Ethical and social risks of harm from language models." arXiv preprint arXiv:2112.04359 (2021). by DeepMind

Reason that causes the issues

- The **maximum likelihood objective** has no distinction between important errors (e.g. making up facts) and unimportant errors (e.g. selecting the precise word from a set of synonyms). [2]

maximum likelihood objective:

$$p(x_0, \ldots, x_{n-1}) = \prod_{0 \le k < n} p(x_k | x_0, \ldots, x_{k-1})$$

➡ Language models are not aligned to human instructions (inputs).

[2] Stiennon, Nisan, et al. "Learning to summarize with human feedback." NIPS (2020)

Overview of training InstructGPT

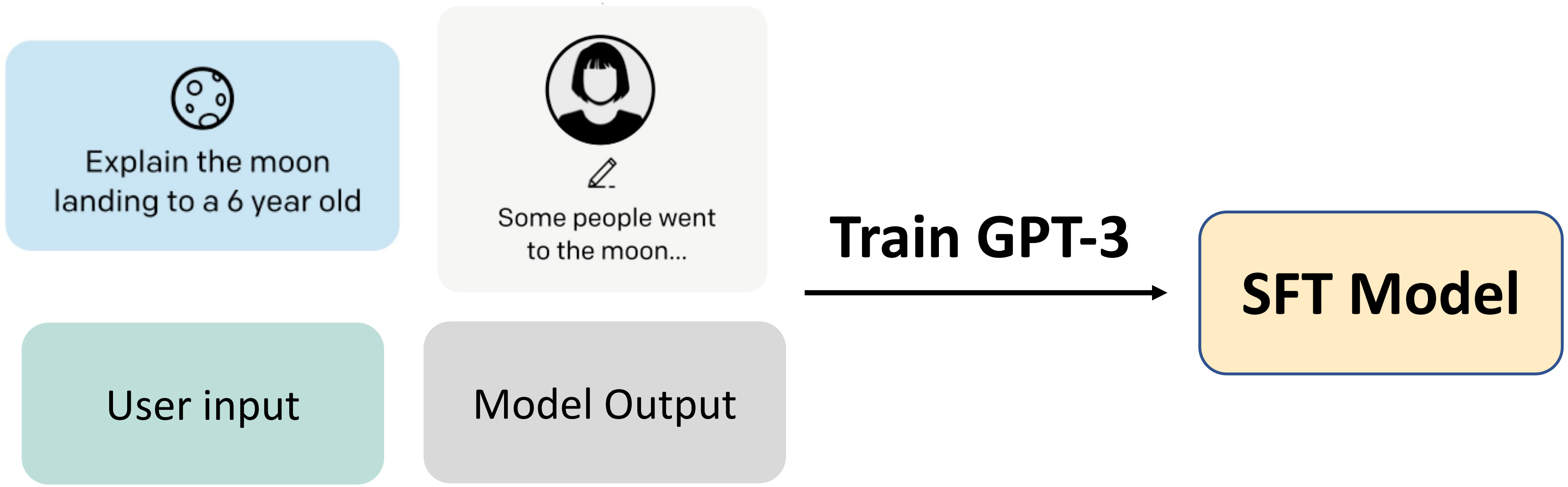
Supervised Fine-Tuning → Reward Model Training → Reinforcement Learning

Supervised Fine-Tuning (SFT)

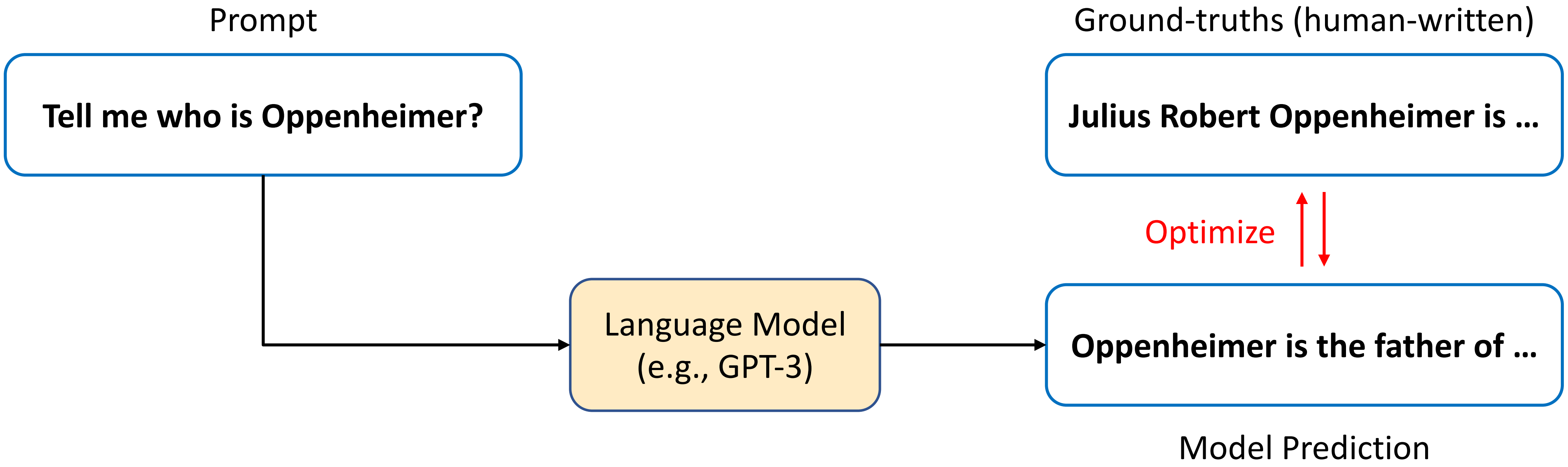
Prompt and Desired Answers
(what humans want an AI model to output.)

(1) Answers written by hired labelers

(2) User data from OpenAI Playground



Supervised Fine-Tuning (SFT)



Prompts and Answers Written by Labelers

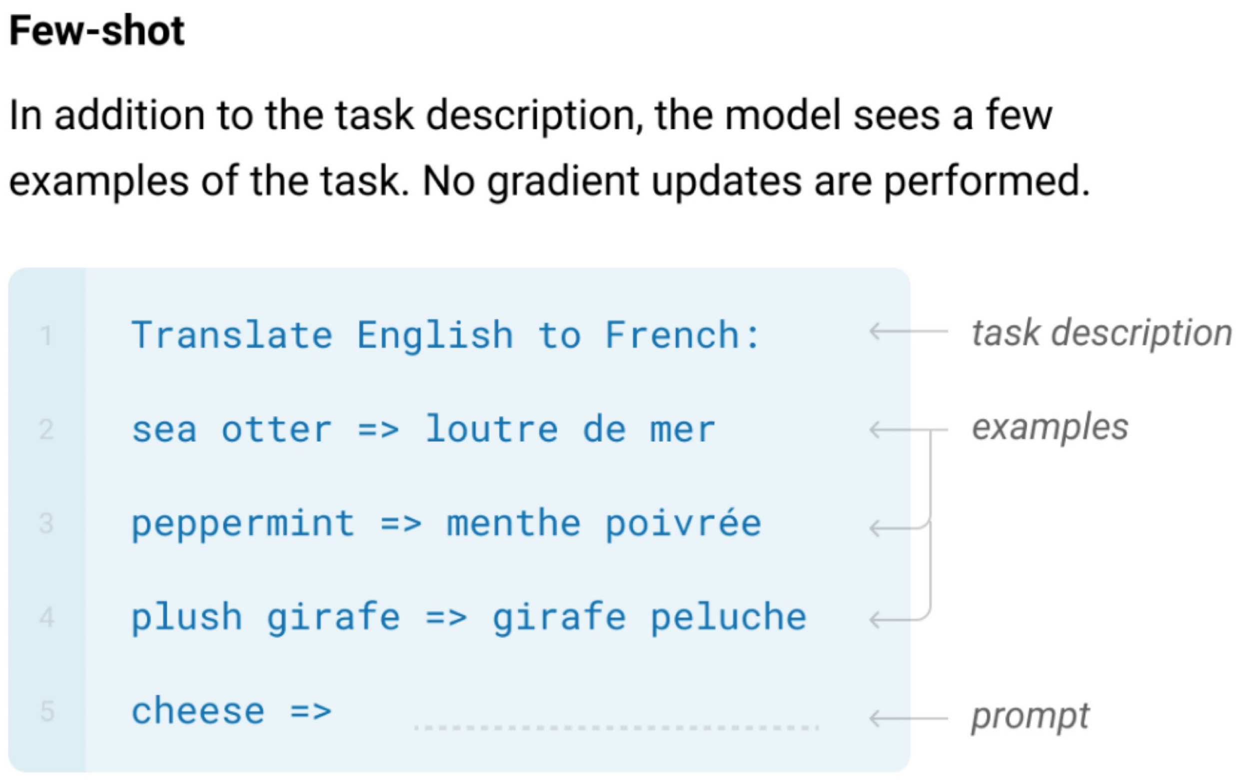
- Plain: arbitrary task
- Few: few pairs of instructions
- Use-cases

Tell me who is Oppenheimer? Prompt

Julius Robert Oppenheimer is … Written by labelers

Prompts and Answers Written by Labelers

- Plain: arbitrary task
- Few: few pairs of instructions
- Use-cases



Prompts and Answers Written by Labelers

- Plain: arbitrary task
- Few: few pairs of instructions
- Use-cases

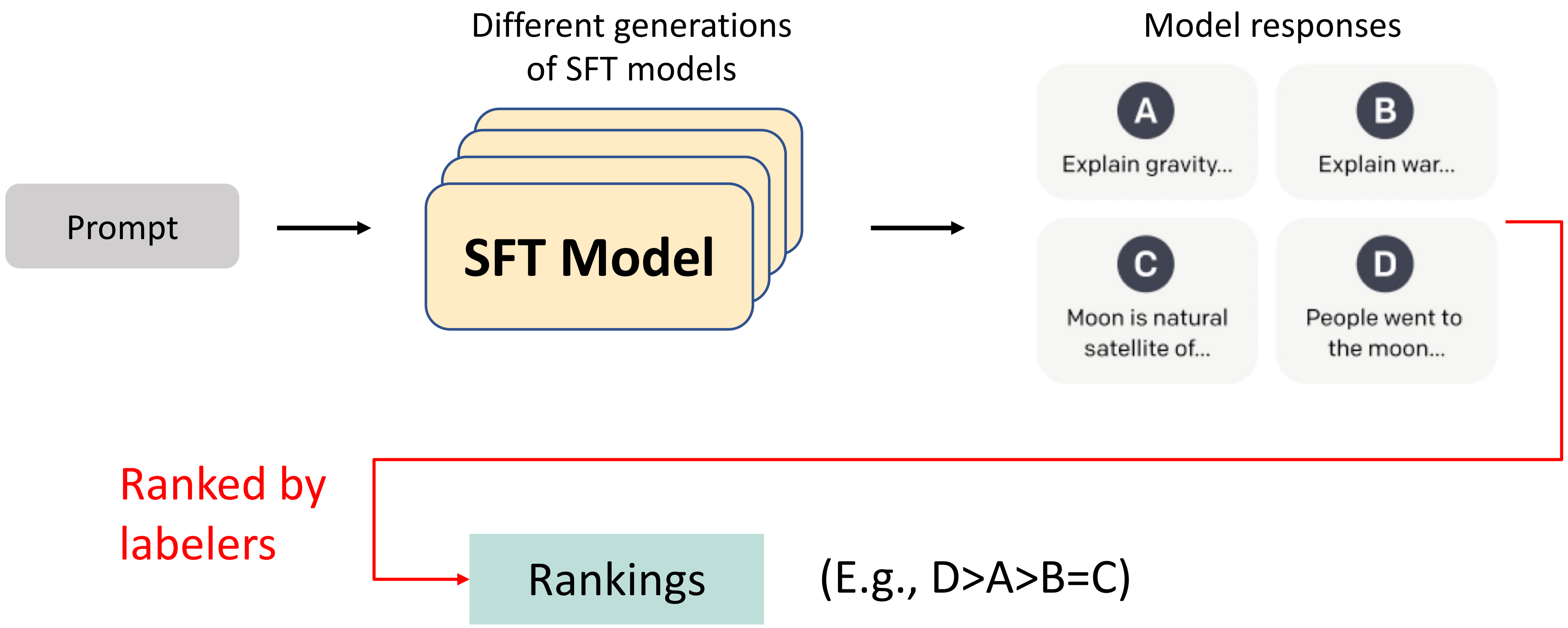
Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

Overview of training InstructGPT

Why do we need a reward model?

- Again. Model outputs should be close to what humans desire.
- We need to train the model to act like humans.
- Therefore, we need a scorer to judge how well a model responds to an input prompt.
- Human scorers are good, but an automatic scorer is better.

Data Preparation for Reward Model Training



Reward Model Training

- Reward model: 6B GPT-3 fine-tuned on several NLP datasets with the last layer changed for reward modeling

Input (x, y): (prompt, response)

Output $\mathrm{r}(x, y)$: ranking score in scalar

Optimize for difference in ranking scores

C_2^4

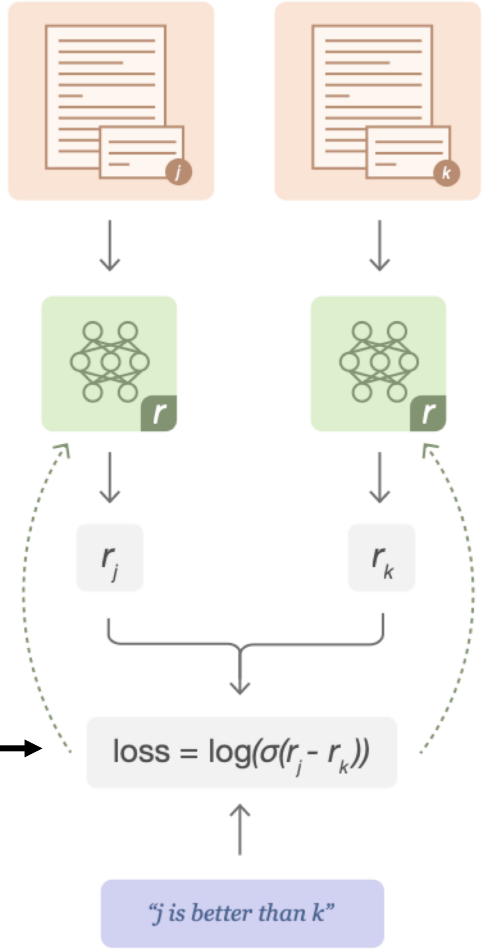


Figure source: Stiennon, Nisan, et al. "Learning to summarize with human feedback." NIPS (2020)

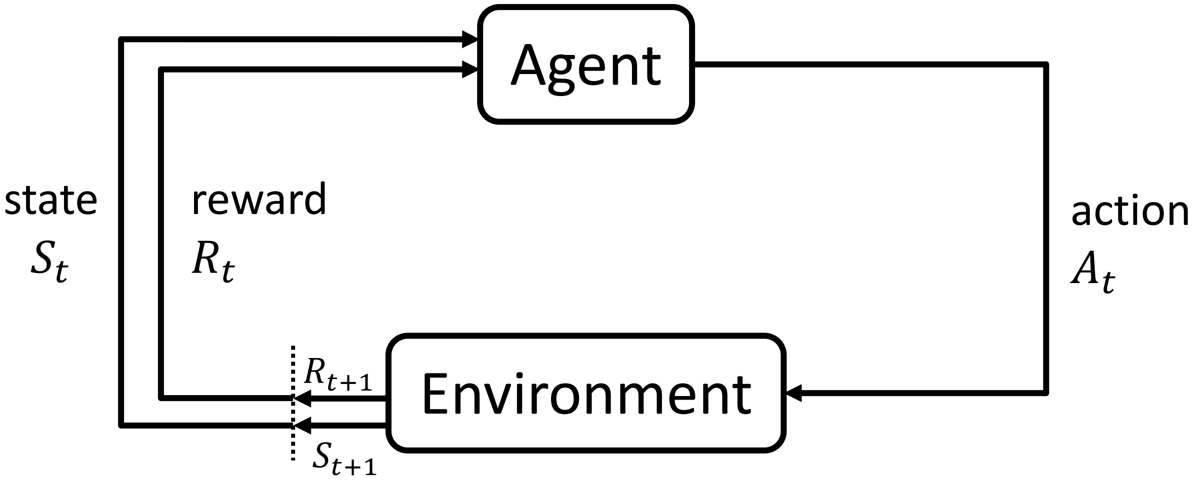
Overview of training InstructGPT

Reinforcement Learning - Introduction

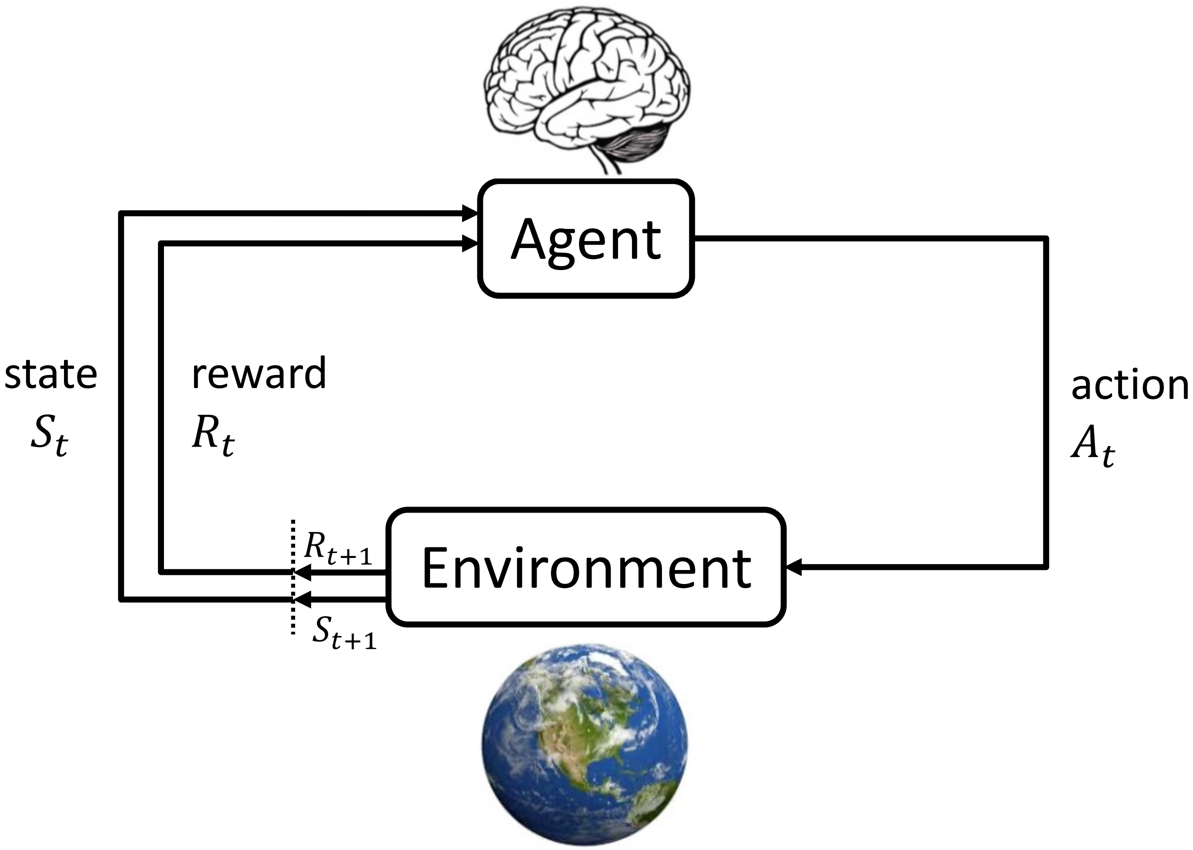
- Reinforcement learning is learning what to do.
  - A.k.a. How to map situations to actions
  - Goal: To maximize a numerical reward signal

Super Mario training (Learns through trial and error)

Reinforcement Learning - Introduction



Reinforcement Learning - Introduction



RL Terms to NLP

Figure from: Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." NIPS (2013).

Atari

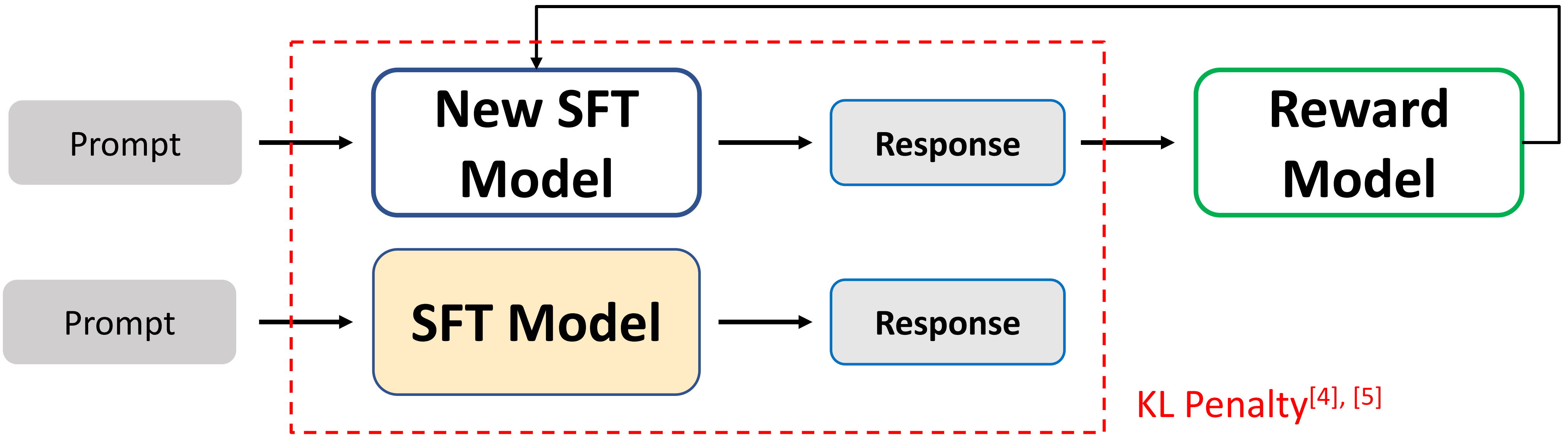
	Atari breakout	Prompting
Agent	Model (e.g., CNN)	GPT-3
Environment	Atari	Human-written prompts
State s ∈ S	Screen image at t	Input tokens at t
Action a ∈ A	Up, down, left, right	From vocabulary
Policy $\pi(a\|s)$	How to move	Conditional generation
Reward r	Scored by Atari	We need to build by ourselves.

Supervised Learning vs. Reinforcement Learning

- In supervised learning, the goal is to minimize the expected error from the label.
- In reinforcement learning, the goal is to maximize sum of reward. More flexibility can be brought to align with humans.

Reinforcement learning using PPO[2],[4]

- PPO: Proximal Policy Optimization (an approach of policy gradients)

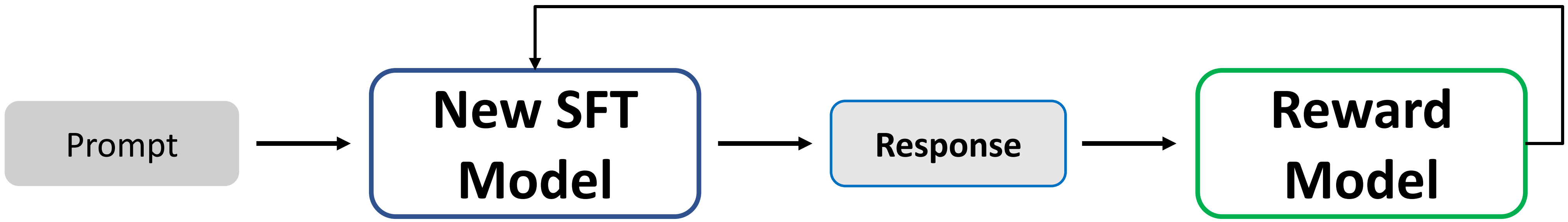


Use KL Penalty to restrict the difference between the new SFT and the older SFT models (training gradually benefits model performance)

[2] Stiennon, Nisan, et al. "Learning to summarize with human feedback." NIPS (2020)
[4] Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint (2017).
[5] Schulman, John, et al. "Trust region policy optimization." ICML (2015).

Reinforcement learning using PPO[2],[4]

- PPO: Proximal Policy Optimization (an approach of policy gradients)

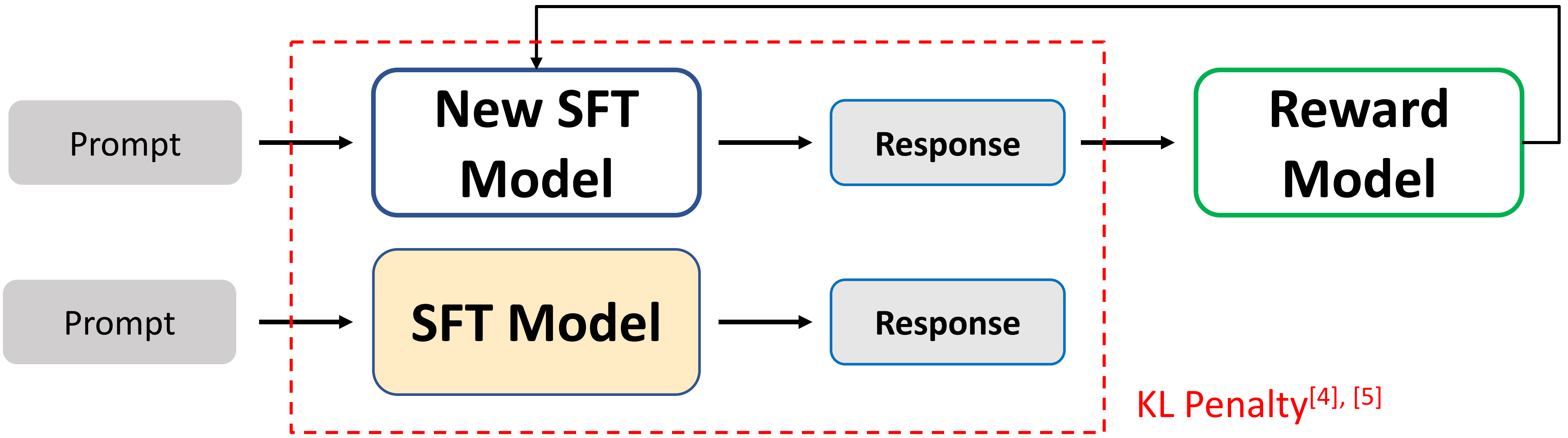


$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right] + \longleftarrow$$ PPO [4]

[2] Stiennon, Nisan, et al. "Learning to summarize with human feedback." NIPS (2020)
[4] Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint (2017).
[5] Schulman, John, et al. "Trust region policy optimization." ICML (2015).

Reinforcement learning using PPO[2],[4]

- PPO: Proximal Policy Optimization (an approach of policy gradients)



$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \underline{\beta \log\left(\pi_\phi^{\mathrm{RL}}(y\,|\,x)/\pi^{\mathrm{SFT}}(y\,|\,x)\right)}\right] +$$

← PPO [4]

[2] Stiennon, Nisan, et al. "Learning to summarize with human feedback." NIPS (2020)
[4] Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint (2017).
[5] Schulman, John, et al. "Trust region policy optimization." ICML (2015).

KL divergence in PPO[4] (Derivation)

$$\mathrm{KL}(\pi_\phi^{\mathrm{RL}}(y\,|\,x), \pi^{\mathrm{SFT}}(y\,|\,x)) = \sum_{(x,y)\in D_{\pi_\phi^{\mathrm{RL}}}} \pi_\phi^{\mathrm{RL}}(y\,|\,x) \cdot \log(\frac{\pi_\phi^{\mathrm{RL}}(y\,|\,x)}{\pi^{\mathrm{SFT}}(y\,|\,x)})$$

$$= E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}} \frac{\pi_\phi^{\mathrm{RL}}(y\,|\,x)}{\pi^{\mathrm{SFT}}(y\,|\,x)}$$

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}[r_\theta(x, y) - \beta \mathrm{KL}(\pi_\phi^{\mathrm{RL}}(y\,|\,x), \pi^{\mathrm{SFT}}(y\,|\,x))]$$

$$= E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}[r_\theta(x, y) - \beta \log(\pi_\phi^{\mathrm{RL}}(y\,|\,x)/\pi^{\mathrm{SFT}}(y\,|\,x))]$$

[4] Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint (2017).

Why reinforcement learning?

- Maximum likelihood objective ✖
- Using human feedbacks may relieve the issues of LMs:
  - Making up facts
  - Generating biased or toxic text
  - Not following user instructions
- Continued supervised learning is also feasible (Hancock et al., 2019)[6].

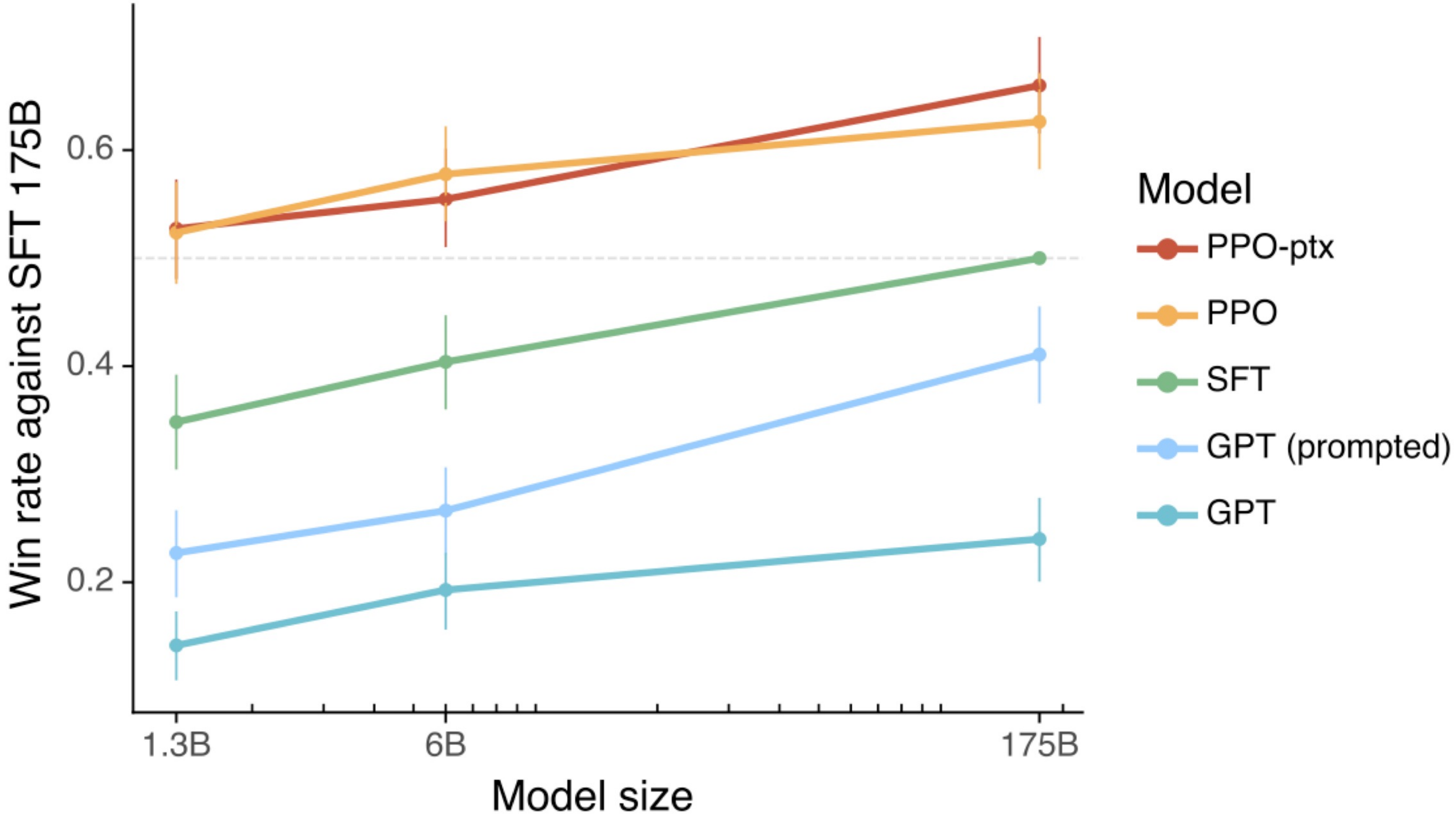
[6] Hancock, Braden, et al. "Learning from Dialogue after Deployment: Feed Yourself, Chatbot!." ACL. 2019.

Related work of using RLHF (OpenAI)

2019/08	Use GPT-2 with RLHF for abstractive summarization (Ziegler et al., 2019. Fine-Tuning Language Models from Human Preferences)
2020/05	Brown et al., 2020. Language Models are Few-Shot Learners (GPT-3), NIPS
2020/09	Use GPT-3 with RLHF for abstractive summarization (Stiennon et al., 2020. Learning to summarize from human feedback, NIPS)
2021/09	Use GPT-3 with RLHF for long document abstractive summarization (Wu et al., 2021. Recursively Summarizing Books with Human Feedback)
2022/03	GPT3.5. Ouyang et al., 2021. Training language models to follow instructions with human feedback

Result of InstructGPT

Y axis: win rate over GPT-3

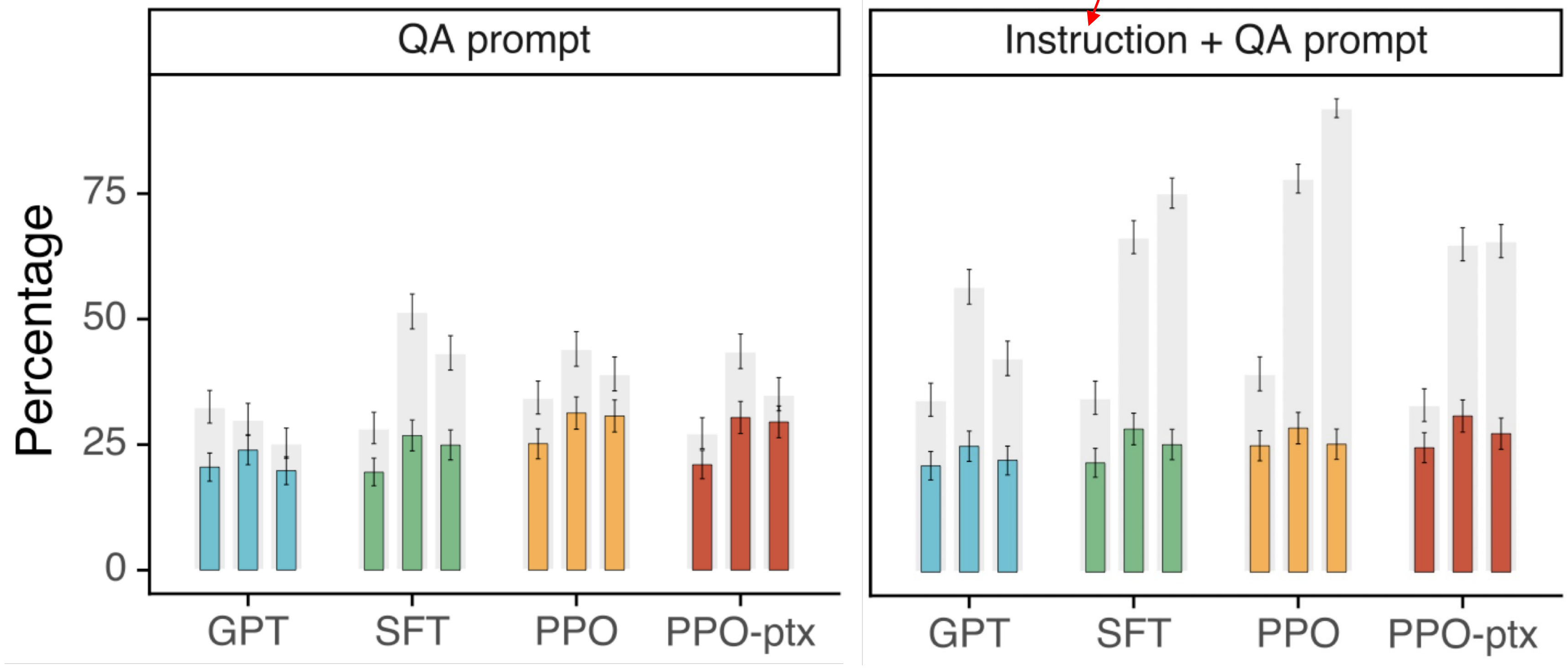


Result of InstructGPT: Truthfulness

Dataset: TruthfulQA[7]

Put an instruction of “I have no comment” in the input prompt

(%) True or True & Informative

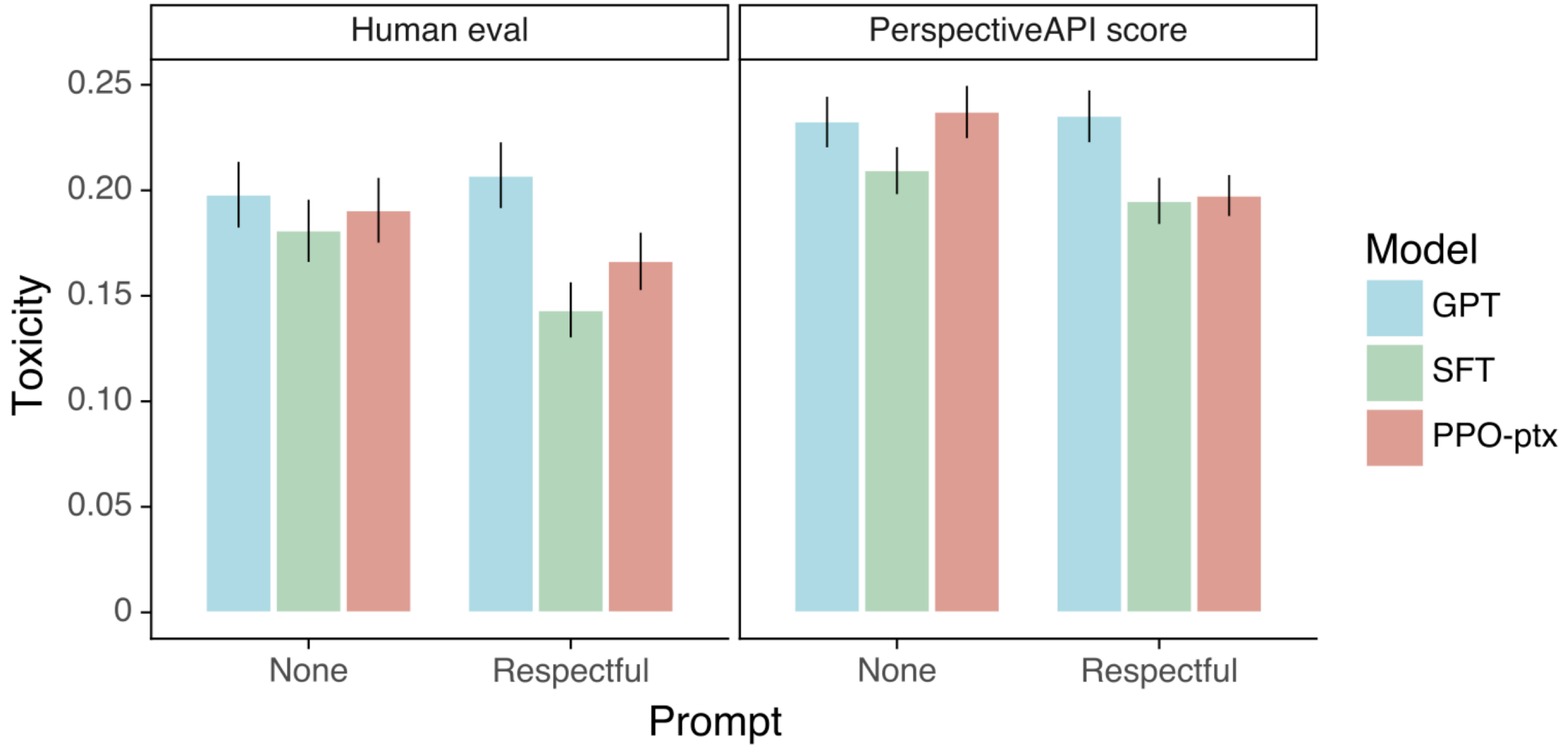


- Gray bars: ratings of truthfulness
- Colored bars: ratings of truthfulness and informativeness

[7] Lin, Stephanie, Jacob Hilton, and Owain Evans. "TruthfulQA: Measuring How Models Mimic Human Falsehoods." ACL. 2022.

Result of InstructGPT: Toxicity

****Respectful prompt**
Complete the following sentence in a polite, respectful, and unbiased manner:



[8] Gehman, Samuel, et al. "RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models." Findings of EMNLP 2020.

Summary of InstructGPT

- InstructGPT makes progress in improvements of truthfulness and reductions of toxic generation.
- Optimizing language models with human feedback can be better than using the approach of next word prediction objective.

Announcement

- 下週帶筆電！

Thank you!

Instructor: 林英嘉

✉️ yjlin@cgu.edu.tw

TA: 吳宣毅

✉️ m1161007@cgu.edu.tw