Natural Language Processing

LLM alignment

Marco Kuhlmann Department of Computer and Information Science



This work is licensed under a <u>Creative Commons Attribution 4.0 International License</u>.

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

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

<u>Aligning language models to follow instructions</u> (OpenAI, 2022)

	unsupervised pre-training	instruction fine-tuning	reward modelling	
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				

reinforcement learning



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.

<u>Wang et al., 2022</u>

Task Instruction Definition "... Given an utterance and recent dialogue context containing past 3 utterances (wherever available), output 'Yes' if the utterance contains the small-talk strategy, otherwise output 'No'. Small-talk is a cooperative negotiation strategy. It is used for discussing topics apart from the negotiation, to build a rapport with the opponent." **Positive Examples** • Input: "Context: ... 'That's fantastic, I'm glad we came to something we both agree with.' Utterance: 'Me too. I hope you have a wonderful camping trip." • Output: "Yes" • Explanation: "The participant engages in small talk when wishing their opponent to have a wonderful trip." **Negative Examples** • Input: "Context: ... 'Sounds good, I need food the most, what is your most needed item?!' Utterance: 'My item is food too'." • Output: "Yes" • Explanation: "The utterance only takes the negotiation forward and there is no side talk. Hence, the correct answer is 'No'." **Evaluation Instances Tk-Instruct** • Input: "Context: ... 'I am excited to spend time with everyone from camp!' Utterance: 'That's awesome! I really love being out here with my son. Do you think you could spare some food?'" • Expected Output: "Yes"





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

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.

Worse

Explain the theory of gravity to a 6 year old.



	unsupervised pre-training	instruction fine-tuning	reward modelling
data	raw text from the Internet billions of words	ideal dialogues 10k–100k	annotated dialogues 100k–1M
	low quality, high quantity	low quantity, high quality	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 mod	el	

reinforcement learning





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:

$$loss(\boldsymbol{\theta}) := -\frac{1}{m} \sum_{i=1}^{m} log(\sigma(R_{\boldsymbol{\theta}}(x_i, y_i^+) - R_{\boldsymbol{\theta}}(x_i, y_i^-))) + R_{\boldsymbol{\theta}}(x_i, y_i^-)$$

$(i_i^-))$

eferred letion

	unsupervised pre-training	instruction fine-tuning	reward modelling
data	raw text from the Internet billions of words	ideal dialogues 10k–100k	annotated dialogues 100k–1M
	low quality, high quantity	low quantity, high quality	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 mod	el	

reinforcement learning

generated dialogues 10k–100k low quantity, high quality

reinforcement learning generate text for maximal reward

1–100 GPUer several days of training time ChatGPT, Claude



Policy gradient

- 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:

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

reward is positive – take gradient steps to maximise probability reward is negative – take gradient steps to minimise probability

<u>Williams (1992); Schulman et al. (2017)</u>

 (x_i)

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.



 \odot

Explain the moon

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



Step 2

Collect comparison data, and train a reward model.



the outputs from best to worst.

This data is used to train our reward model.



D>G>A=B

 \odot Explain the moon landing to a 6 year old A B Explain gravity. Explain wer..

C O Moon is natural actellite of ...

People went to the moon...

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Figure 2 from <u>Ouyang et al. (2022)</u>

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.



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:

$$R'(x, y) = R(x, y) - \frac{\beta \log[p_{\theta}^{RL}(y | x)/p^{FT}(y | x)]}{\beta \log[p_{\theta}^{RL}(y | x)/p^{FT}(y | x)]}$$
sample reward model penalty based on KL divergence

|x|

Effectiveness of human feedback



12.9B

Stiennon (2020)