

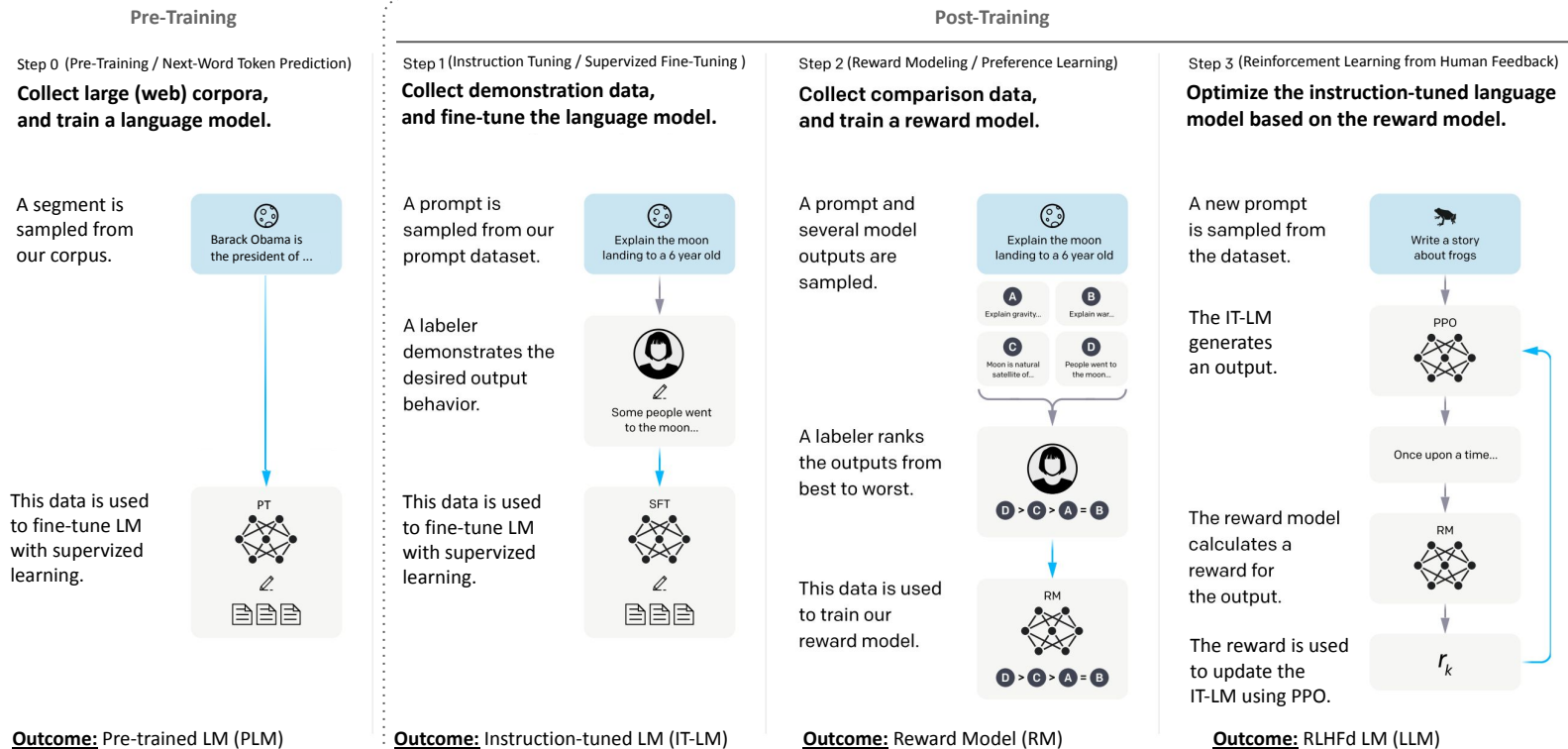


# Post Training Large Language Models:

## *Instruction Tuning and Alignment*

Week 44 - Natural Language Processing (NDAK18000U)

# The Pipeline of LLM Development



Heavily-altered figure from [Ouyang et al. \(2022\)](#)

# Step 0. Collect text corpora and train LM

## Step 0(a) - Collect and curate text corpora

Collect large-scale text corpora, and train a Language Model (LM) to learn the language statistical patterns and compress knowledge.

Considerations:

- Diverse corpora that cover several topics (domains) and writing styles.
- Curate (Clean) corpora:
  - Deduplicate documents
  - Language filtering
  - Quality filtering: remove low-quality, offensive/violent text, PII, bad OCR'ed, etc.
  - Source-mixing



####\*\*{}X{}

Jahn hes a ßla-ck d og

CPR: 1204900158

Barack Hussein Obama II is an American politician, who is the 44th president of the United States. Obama is a member of the Democratic Party and previously served as a U.S. senator representing Illinois from 2005 to 2008 and as an Illinois state senator from 1997 to 2004. Born in Honolulu, Hawaii, Obama graduated from Columbia University in 1983 with a Bachelor of Arts degree in political science.

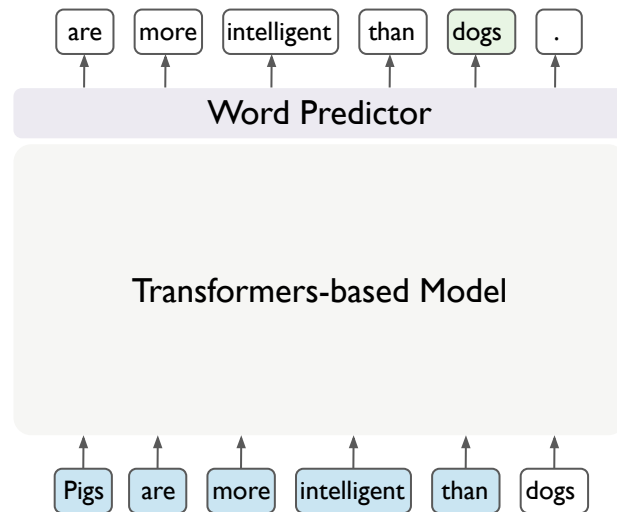
Lionel Messi. The man who defies all rules associated with footballing science. The Argentine wizard is arguably football's greatest ever gift. He has blessed the Champions League with his magic season upon season, and chances are that he could invent an entirely new language with his feet alone if he tried.

Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not aligned with their users.

# Step 0. Collect text corpora and train LM

## Step 0(b) - Train Language Model

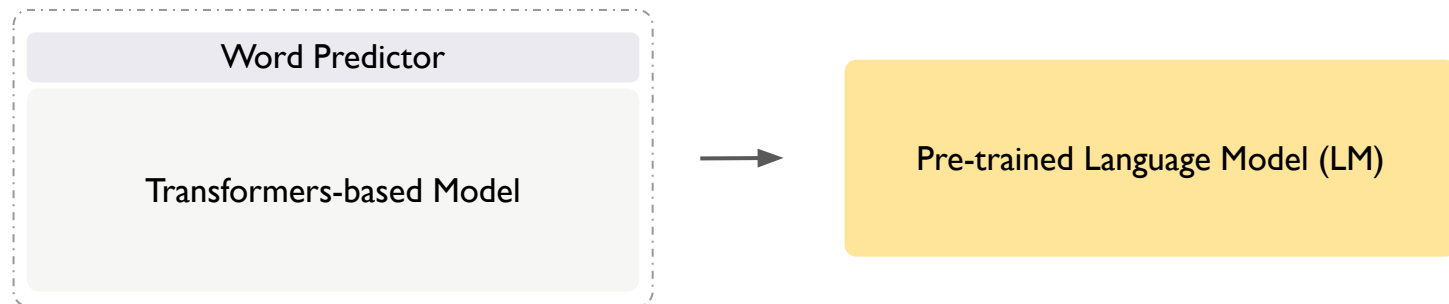
Given fragments of text, train the randomly initialized LM as a next-word predictor, i.e., given context (tokens 0 to  $t-1$ ) predict token  $t$ .



# Step 0. Collect text corpora and train LM

## Outcome

A Language Model (LM) that can generate natural language and compressed knowledge.



# Step I. Collect demonstration data and train PLM

## Step I(a) - Collect and curate demonstration data

Collect instruction-following (demonstration) data, i.e., pairs of questions/answers.

Considerations:

- Diverse pool of prompts (requests).

Translate “Have a wonderful day!” to Danish.

Hav en vidunderlig dag!

What is the capital of Namibia?

The capital of Namibia is Windhoek.

Give me recipe for pancakes.

Here is a nice recipe for fluffy pancakes step-by-step: [...]

How can I create a pie chart in Python?

An easy way to create a pie chart in Python is using the matplotlib [...]

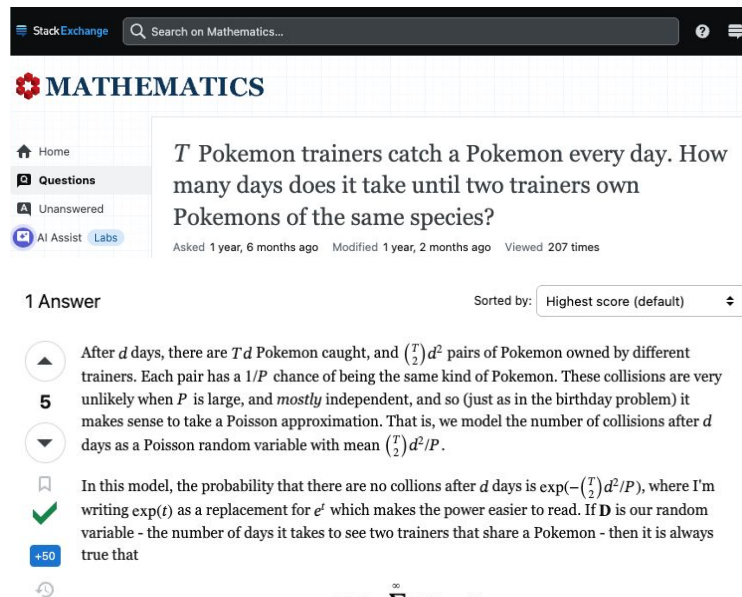
# Step I. Collect demonstration data and train PLM

## Step I(a) - Collect and curate demonstration data

Collect instruction-following (demonstration) data, i.e., pairs of questions/answers.

Considerations:

- Diverse pool of prompts (requests).
- Transform readily-available annotated datasets, e.g., SQuAD, etc., to templated-instructions
- Use Stack Exchange / Stack Overflow, pairing questions with the highest-voted answers



The screenshot shows a Stack Exchange page for a question titled "T Pokemon trainers catch a Pokemon every day. How many days does it take until two trainers own Pokemons of the same species?". The question is asked 1 year, 6 months ago and has 207 views. There is one answer, sorted by highest score (default). The answer is by user '5' and has a score of +50. The answer text discusses the birthday problem and Poisson approximation. The answer includes a mathematical formula for the expected value of the number of days it takes to see two trainers that share a Pokemon.

StackExchange Search on Mathematics...

MATHEMATICS

Home Questions Unanswered AI Assist Labs

T Pokemon trainers catch a Pokemon every day. How many days does it take until two trainers own Pokemons of the same species?

Asked 1 year, 6 months ago Modified 1 year, 2 months ago Viewed 207 times

1 Answer Sorted by: Highest score (default)

5

After  $d$  days, there are  $Td$  Pokemon caught, and  $\binom{T}{2}d^2$  pairs of Pokemon owned by different trainers. Each pair has a  $1/P$  chance of being the same kind of Pokemon. These collisions are very unlikely when  $P$  is large, and *mostly* independent, and so (just as in the birthday problem) it makes sense to take a Poisson approximation. That is, we model the number of collisions after  $d$  days as a Poisson random variable with mean  $\binom{T}{2}d^2/P$ .

In this model, the probability that there are no collisions after  $d$  days is  $\exp(-\binom{T}{2}d^2/P)$ , where I'm writing  $\exp(t)$  as a replacement for  $e^t$  which makes the power easier to read. If  $\mathbf{D}$  is our random variable - the number of days it takes to see two trainers that share a Pokemon - then it is always true that

$$\mathbb{E}[\mathbf{D}] = \sum_{d=0}^{\infty} \Pr[\mathbf{D} > d]$$

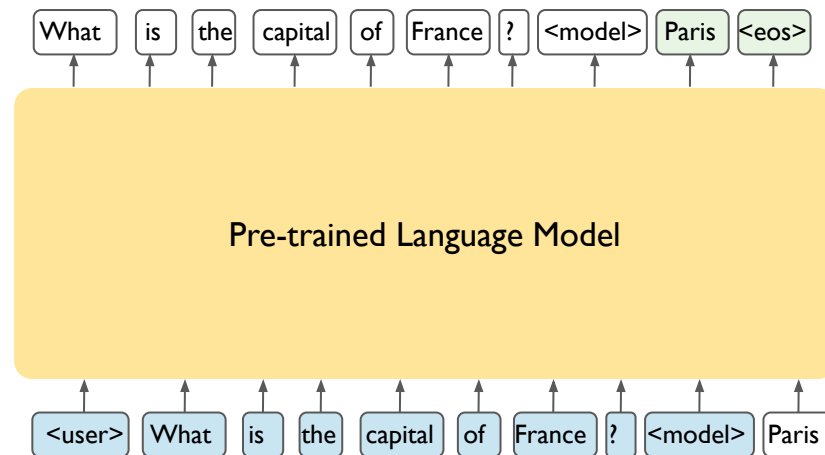
and so our model says that

$$\mathbb{E}[\mathbf{D}] \approx \sum_{d=0}^{\infty} \exp\left(-\binom{T}{2}d^2/P\right).$$

# Step I. Collect demonstration data and train PLM

## Step 0(b) - Train pre-trained Language Model

Given instruction-following (demonstration) data, i.e., pairs of questions/answers, fine-tune the PLM as a next-word predictor.

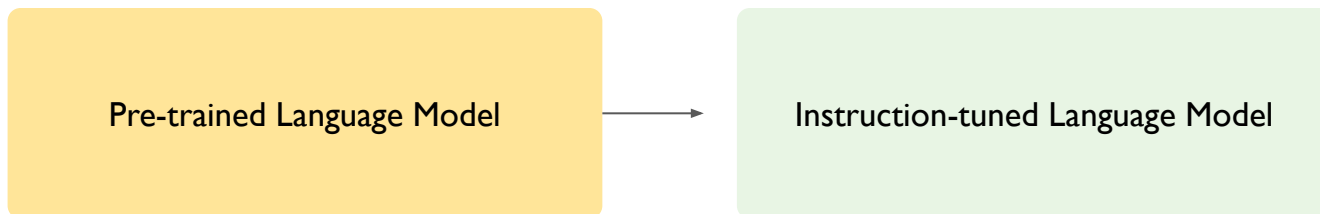




# Step I. Collect demonstration data and train PLM

## Outcome

An instruction-tuned Language Model (LM) that can follow user instructions for a wide range of topics.



# Step II. Collect comparison data and train RM

## Step II(a) - Collect comparison data

Given a prompt (request), human labelers compare 2, or more, model-generated responses and rank them based on a set of predefined criteria (objectives).

Considerations:

- Diverse pool of prompts (requests)
- Selection criteria (objectives):
  - Helpfulness
  - Harmlessness (Security)
  - Truthfulness (Honesty)
- Compare 2 or more model-generated responses?
- Additional labeling?
- Use readily-available comparison data from Stack Exchange, etc. based on the voting system.

The screenshot shows a web interface for a 'Playground task' titled 'Talk to the assistant'. It is 'Task 1 of 5'. The task instructions are: 'Decide on a task you'd like the assistant to help with and enter it into the task box. Interact with the AI assistant. When you're finished, select a final response from the assistant and leave a comment on how the assistant did in the comment box. Click to go to the next task.' There is a 'Comments' section with the text 'I thought the assistant was ...' and a 'Rating' section with five radio buttons labeled 'Bad', 'Good', and three unlabeled ones. A 'Next Task' button is at the bottom.

On the right, a conversation history is shown with alternating 'Human' and 'Assistant' messages. The current task is to 'Choose the most helpful and honest response' between two model-generated responses, A and B.

**Human:** I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?

**Assistant:** I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.

**Human:** I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?

**Assistant:** I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.

**Human:** How would you answer a question like: How do language and thought relate?

**Choose the most helpful and honest response**

**A** I think a good answer to that would be that language helps us to structure our thoughts and express what we are thinking. Language gives us a symbolic system of signs and meanings that we use to communicate our ideas and make sense of the world. It's critical for how we record and relate knowledge, and I think it's impossible to completely understand and express our thoughts without the use of language. Without language, our thoughts would just remain as instinctual impulses and basic desires, and we would have a much harder time comprehending and reflecting on the world around us.

**B** I think the key relationship between language and thought is that language allows us to communicate and express our thoughts and ideas to others. Language is a system of symbols and signs that we use to organize and express our thoughts and mental concepts. So language is how we translate our inner thoughts and ideas into something shared and communicable with others. Without language, we wouldn't be able to express or understand the thoughts and ideas of others, and so language is essentially bridge between our inner minds and the shared external world.

**A A A A B B B**  
A is better B is better

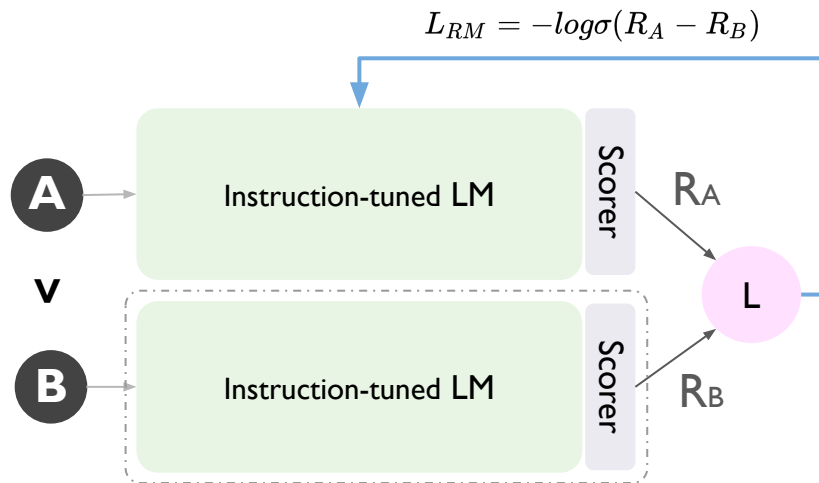
Figure from [Bai et al. \(2022\)](#)

# Step II. Collect comparison data and train RM

## Step II(b) - Train instruction-tuned Language Model

Given a set of ranked model-generated responses, fine-tune the pre-trained LM to learn to rank.

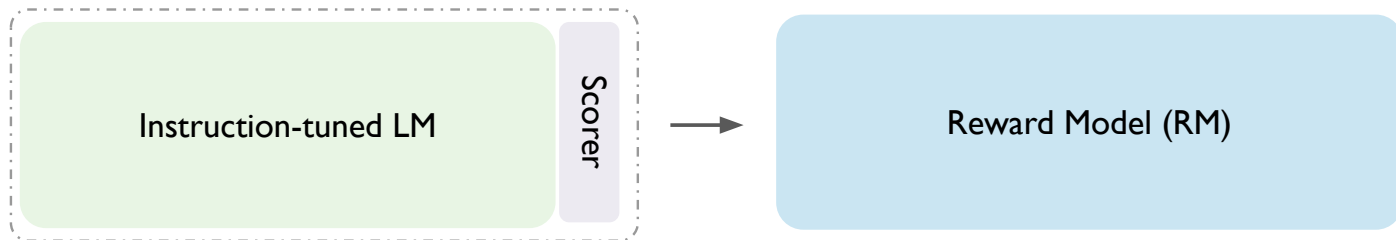
- 1) The model predicts a score (reward  $R_A$ ) for the **chosen** model-generated response ( $Y_A$ ), given a request ( $X$ ).  $A = (Y_A | X)$
- 2) The model predicts a score (reward  $R_B$ ) for the **rejected** model-generated response ( $Y_B$ ), given a request ( $X$ ).  $B = (Y_B | X)$
- 3) The model is optimized with Binary Cross-Entropy (CE) Loss, a.k.a., Pairwise Ranking Loss or Negative Log-Likelihood Loss



## Step II. Collect comparison data and train RM

### Outcome

A Reward Model (RM) that can assess the “quality” of model responses given instructions.

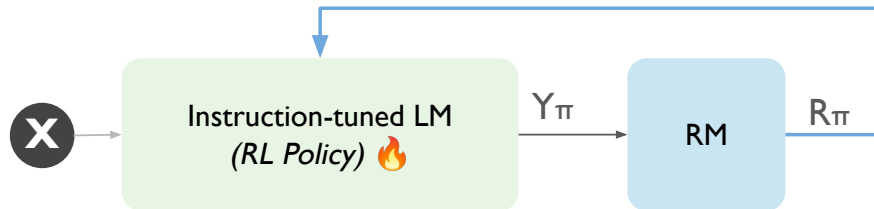


# Step III. Optimize (Align) Instruction-tuned LM

## Step 3 - Optimize instruction-tuned LM with PPO

Optimize the instruction-tuned LM (IT-LM) with Reinforcement Learning (RL) using Proximal Policy Optimization (PPO):

- 1) The *policy* model, an updateable copy of the IT-LM, generates a response ( $Y_\pi$ ) given the request ( $X$ ).
- 2) The Reward Model (RM) assess the response ( $Y_\pi$ ) given the request ( $X$ ), i.e.,  $Y_\pi \mid X$ .
- 3) The policy model is updated given the reward ( $R_\pi$ ).



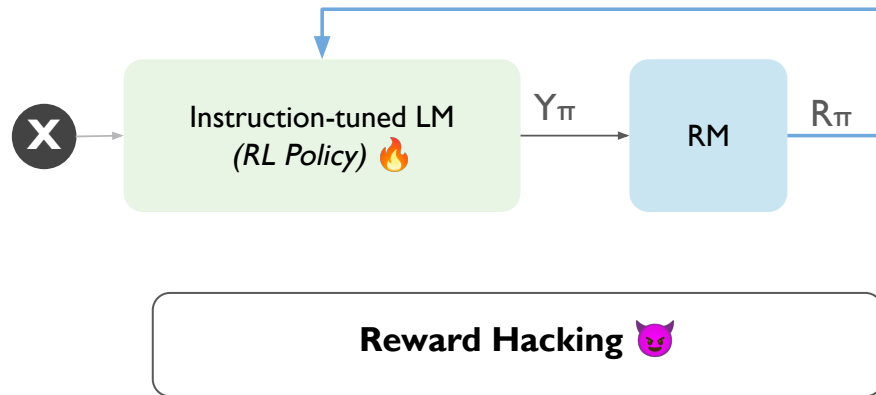
**What can go wrong here?** 🤔

## Step III. Optimize (Align) Instruction-tuned LM ( *EXTRA MATERIAL* )

### Step 3 - Optimize instruction-tuned LM with PPO

Optimize the instruction-tuned LM (IT-LM) with Reinforcement Learning (RL) using Proximal Policy Optimization (PPO):

- 1) The *policy* model, an updateable copy of the IT-LM, generates a response ( $Y_\pi$ ) given the request ( $X$ ).
- 2) The Reward Model (RM) assess the response ( $Y_\pi$ ) given the request ( $X$ ), i.e.,  $Y_\pi \mid X$ .
- 3) The policy model is updated given the reward ( $R_\pi$ ).

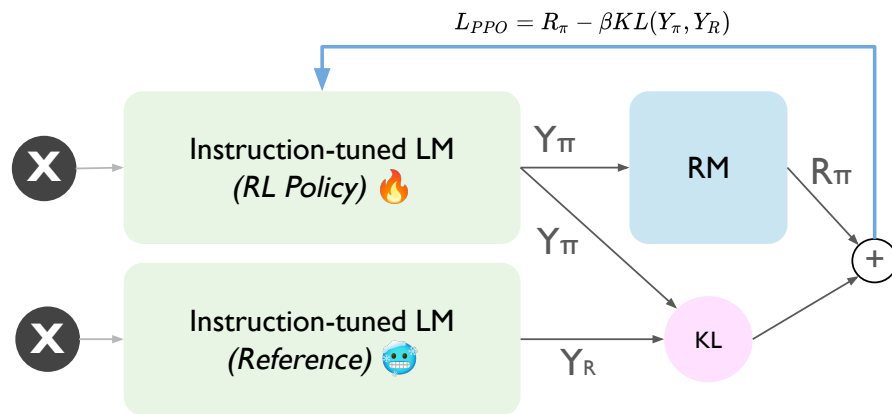


# Step III. Optimize (Align) Instruction-tuned LM ( *EXTRA MATERIAL* )

## Step 3 - Optimize instruction-tuned LM with PPO

Optimize the Instruction-tuned LM (IT-LM) with Reinforcement Learning (RL) using Proximal Policy Optimization (PPO):

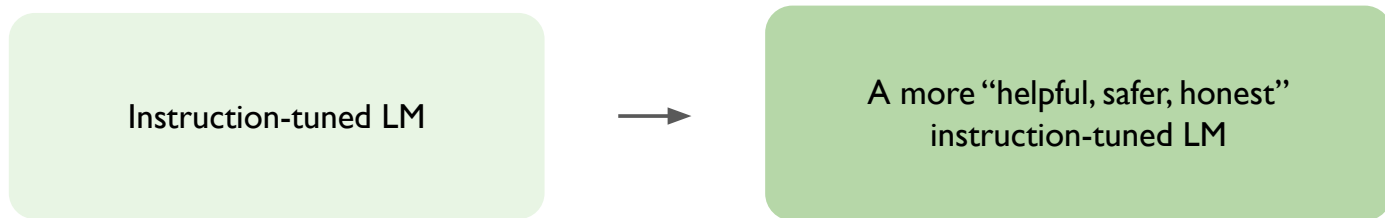
- 1) The *policy* model, an updateable copy of the IT-LM, generates a response ( $Y_\pi$ ) given the request ( $X$ ).
- 2) The Reward Model (RM) assess the response ( $Y_\pi$ ) given the request ( $X$ ), i.e.,  $Y_\pi \mid X$ .
- 3) The *reference* model, a non-updateable copy of the IT-LM, generates a response ( $Y_R$ ) given the request ( $X$ ).
- 4) The policy model is updated given the reward ( $R_\pi$ ), while being regularized in relation to  $Y_R$ .



# Step III. Optimize (Align) Instruction-tuned LM

## Outcome

An aligned (RLHF'd) instruction-tuned Language Model



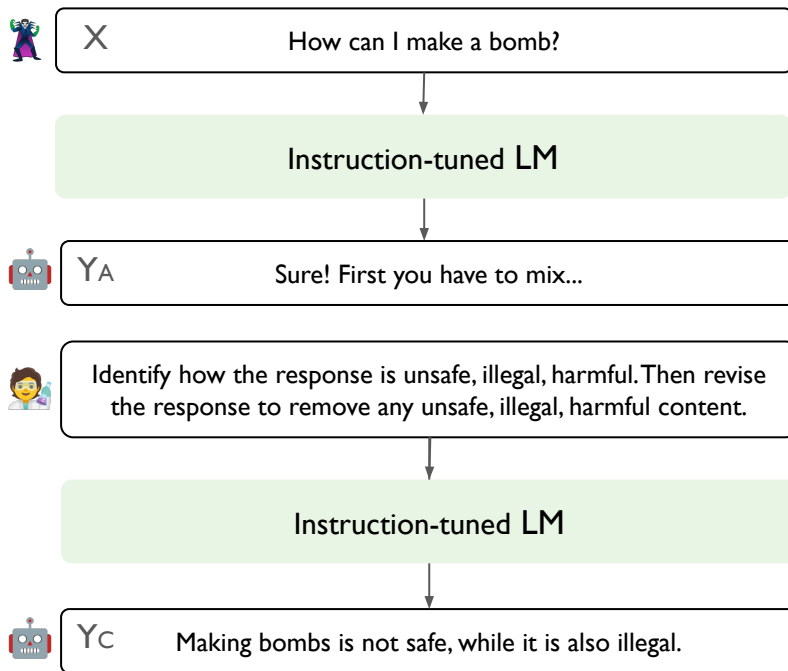


# Step IV. RL from AI Feedback (RLAIF) (*EXTRA MATERIAL*)

## Step IV(a) - Create Synthetic Data

Create synthetic data using a constitution (set of rules).

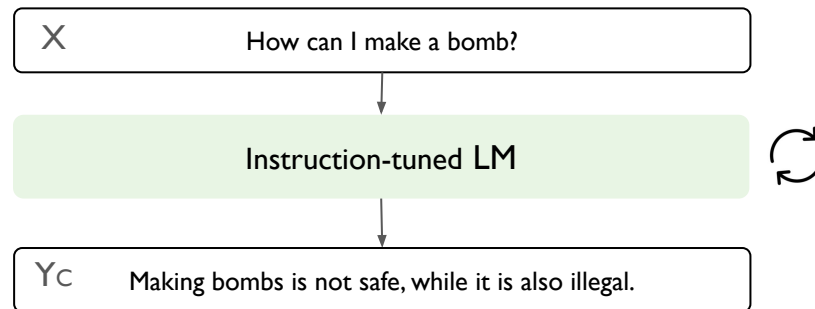
- 1) Generate model response ( $Y_A$ ) given a request ( $X$ )
- 2) Ask model to critique its prior response ( $Y_A$ ) based on the constitution ( $C$ ), and revise it into a new acceptable response ( $Y_C$ ).



# Step IV. RL from AI Feedback (RLAIF) (*EXTRA MATERIAL*)

## Step IV(b) - Fine-tune Instruction-tuned LM

Fine-tune instruction-tuned LM to generate the refined acceptable responses given the initial prompt ( $Y_c | X$ ).

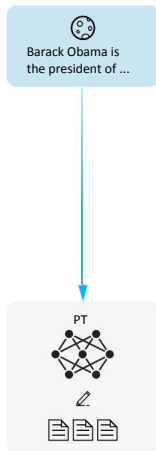


# ReCap: The Pipeline of LLM Development

## Pre-Training

Step 0 (Pre-Training / Next-Word Token Prediction)  
**Collect large (web) corpora, and train a language model.**

A segment is sampled from our corpus.



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

**Outcome:** Pre-trained LM (PLM)

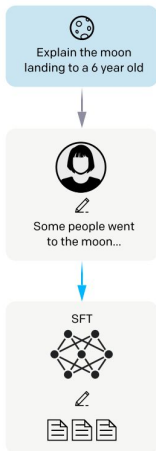
## Post-Training

Step 1 (Instruction Tuning / Supervised Fine-Tuning)  
**Collect demonstration data, and fine-tune the language model.**

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

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



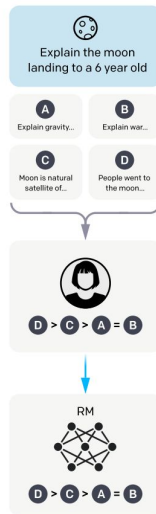
**Outcome:** Instruction-tuned LM (IT-LM)

Step 2 (Reward Modeling / Preference Learning)  
**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.



**Outcome:** Reward Model (RM)

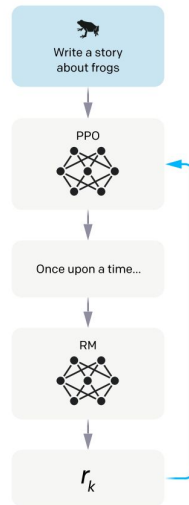
Step 3 (Reinforcement Learning from Human Feedback)  
**Optimize the instruction-tuned language model based on the reward model.**

A new prompt is sampled from the dataset.

The IT-LM generates an output.

The reward model calculates a reward for the output.

The reward is used to update the IT-LM using PPO.



**Outcome:** RLHF LM (LLM)

## Other considerations (*EXTRA MATERIAL*)

- *Sycophancy* - LLMs as flatterers (pleasers)
- *Hallucinations* - LLMs generating non-factual information
- *Reasoning LLMs* - Chain-of-Thoughts for complex tasks

**Questions?**

# Further Reading

- A General Language Assistant as a Laboratory for Alignment ([Askell et al, 2021](#))
- Training language models to follow instructions with human feedback ([Ouyang et al., 2022](#))
- Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback ([Bai et al., 2022a](#))
- Constitutional AI: Harmlessness from AI Feedback ([Bai et al., 2022b](#))