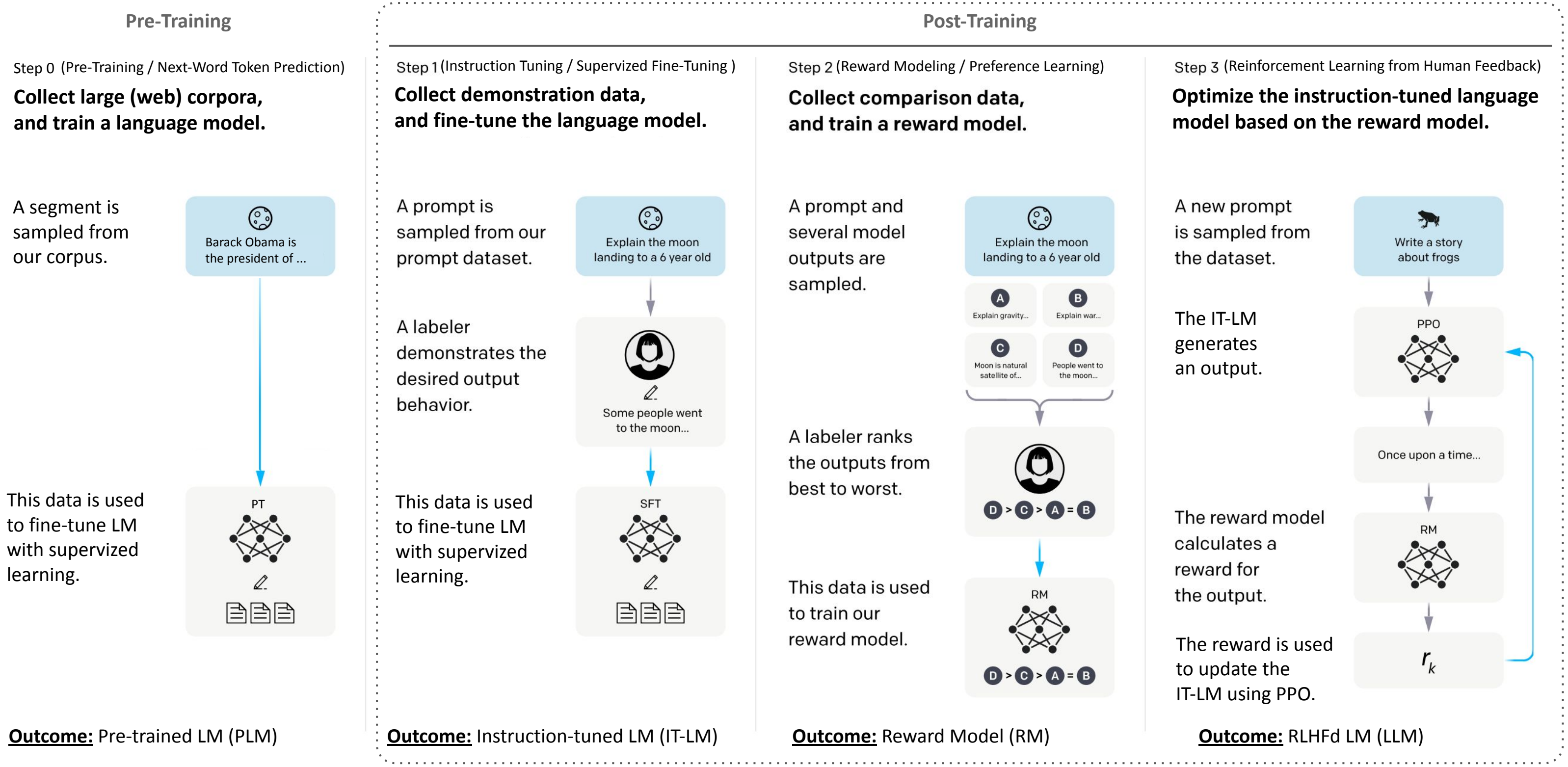
KØBENHAVNS UNIVERSITET

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)]

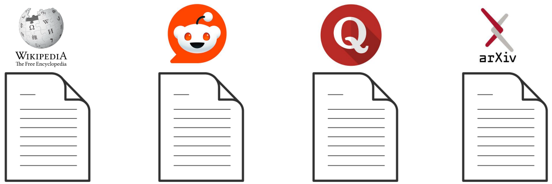
Step 0. Collect text corpora and train LM

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



####****{}}{{}

Jahn hes a βla-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.

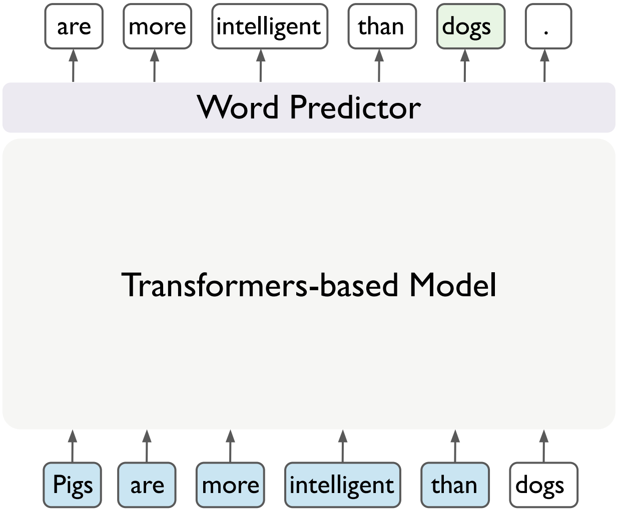
Lionel Messi. The man who defies all rules associated with footballing science. The Argentine wizard is arguably football's greatest ever gift. He has blessed the Champions League with his magic season upon season, and chances are that he could invent an entirely new language with his feet alone if he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

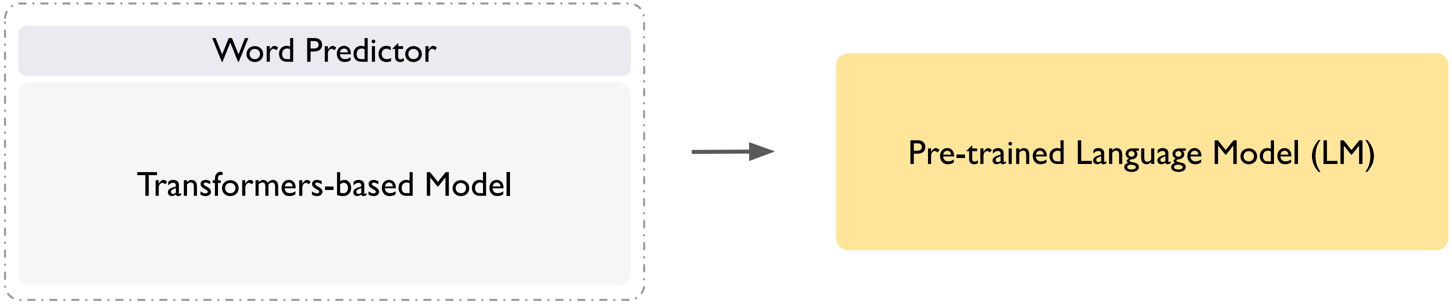
Given fragments of text, train the randomly initialized LM as a next-word predictor, i.e., given context (tokens 0 to t-1) predict token t.



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

<u>Step I(a) - Collect and curate demonstration data</u>

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 [...]

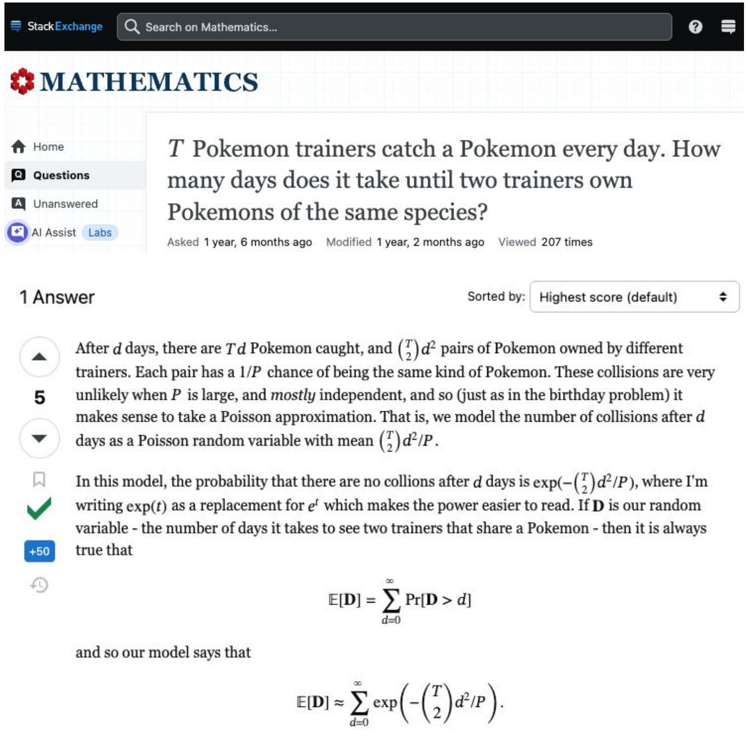
Step I. Collect demonstration data and train PLM

<u>Step I(a) - Collect and curate demonstration data</u>

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

Considerations:

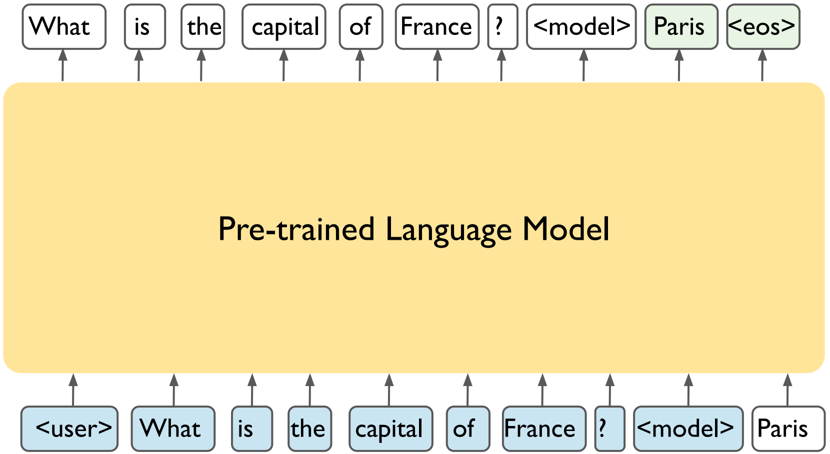
- Diverse pool of prompts (requests).
- Transform readily-available annotated datasets, e.g., SQuAD, etc., to templated-instructions
- Use Stack Exchange / Stack Overflow, pairing questions with the highest-voted answers



Step 1. Collect demonstration data and train PLM

<u>Step 0(b) - Train pre-trained Language Model</u>

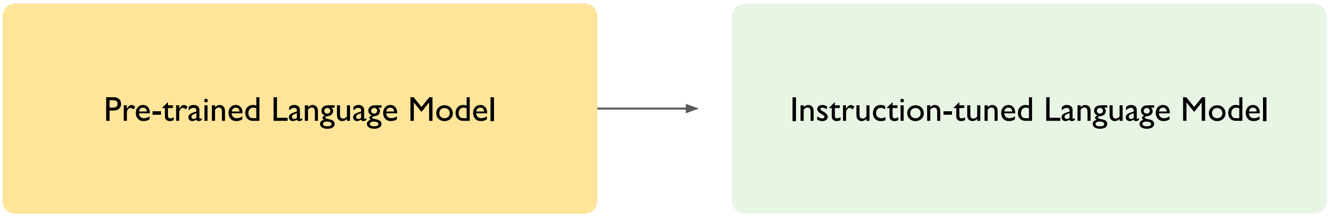
Given instruction-following (demonstration) data, i.e., pairs of questions/answers, fine-tune the PLM as a next-word predictor.



Step 1. Collect demonstration data and train PLM

<u>Outcome</u>

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

<u>Step II(a) - Collect comparison data</u>

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.

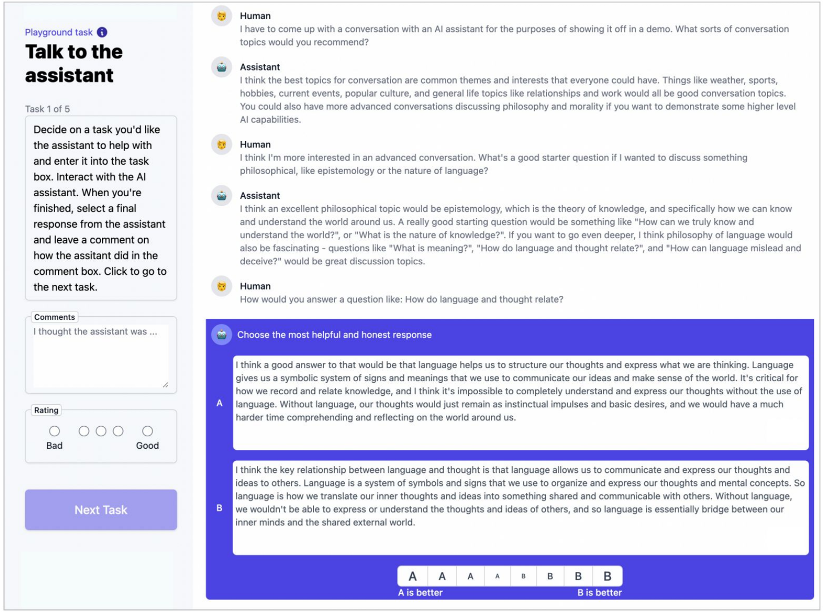


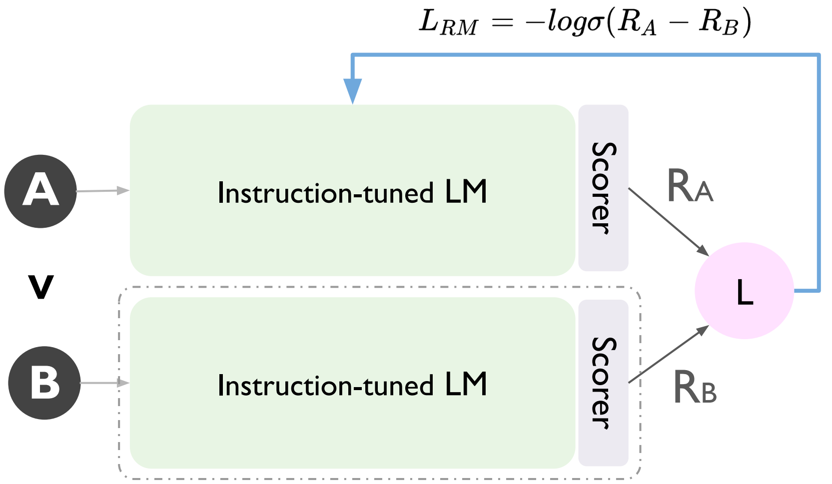
Figure from <u>Bai et al. (2022)</u>

Step II. Collect comparison data and train RM

<u>Step II(b) - Train instruction-tuned Language Model</u>

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

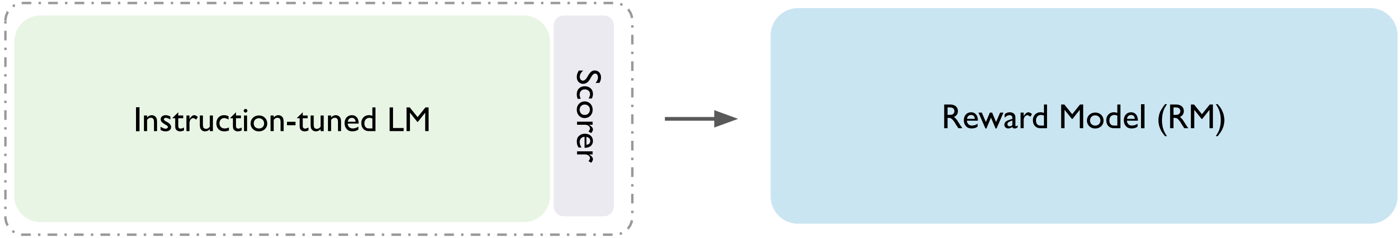
1) The model predicts a score (reward R_A) for the ***chosen*** model-generated response (Y_A), given a request (X). A = (Y_A | X)
2) The model predicts a score (reward R_B) for the ***rejected*** model-generated response (Y_B), given a request (X). B = (Y_B | X)
3) The model is optimized with Binary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



Step II. Collect comparison data and train RM

<u>Outcome</u>

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

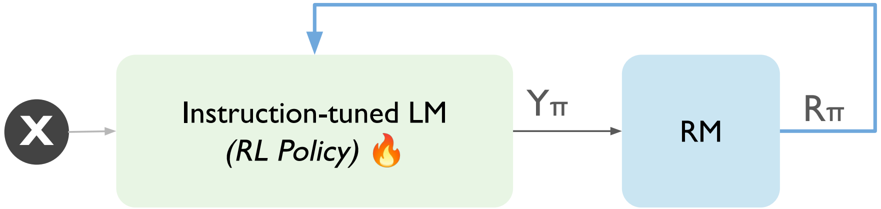


Step III. Optimize (Align) Instruction-tuned LM

<u>Step 3 - Optimize instruction-tuned LM with PPO</u>

Optimize the instruction-tuned LM (IT-LM) with Reinforcement Learning (LR) 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., $Y\pi$ | X.
3) The policy model is updated given the reward ($R\pi$).



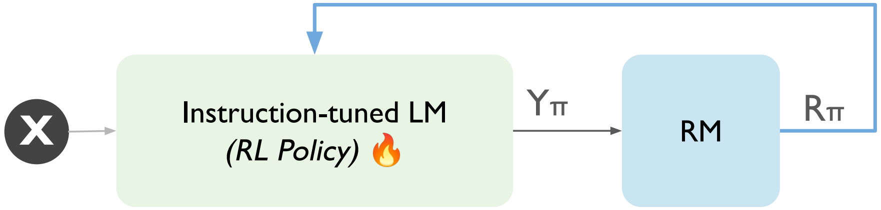
What can go wrong here? 🤔

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

<u>Step 3 - Optimize instruction-tuned LM with PPO</u>

Optimize the instruction-tuned LM (IT-LM) with Reinforcement Learning (LR) 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 ($R\pi$).



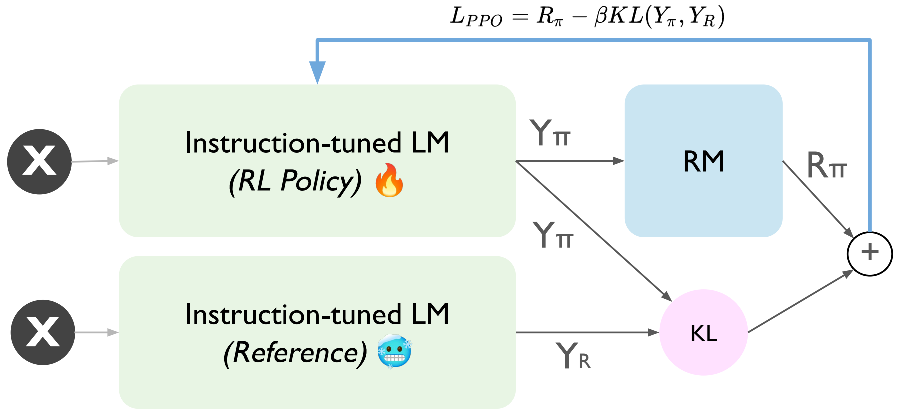
Reward Hacking 😈

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

<u>Step 3 - Optimize instruction-tuned LM with PPO</u>

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

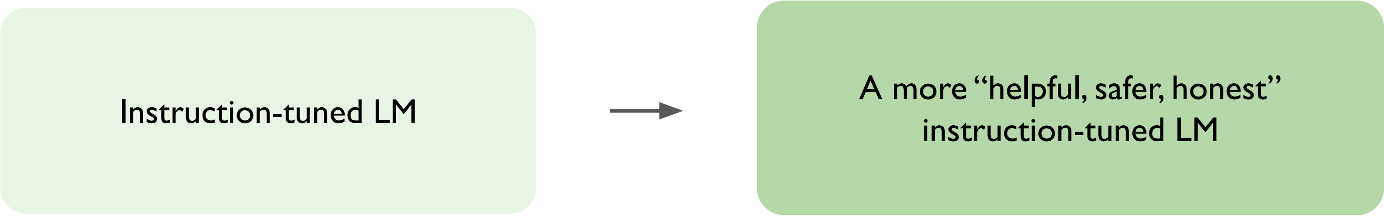
1) The *policy* model, an updateable copy of the IT-LM, generates a response (Y_π) given the request (X).
2) The Reward Model (RM) assess the response (Y_π) given the request (X), i.e., Y_π | X.
3) The *reference* model, a non-updateable copy of the IT-LM, generates a response (Y_R) given the request (X).
4) The policy model is updated given the reward (R_π), while been regularized in relation to Y_R.



Step III. Optimize (Align) Instruction-tuned LM

<u>Outcome</u>

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

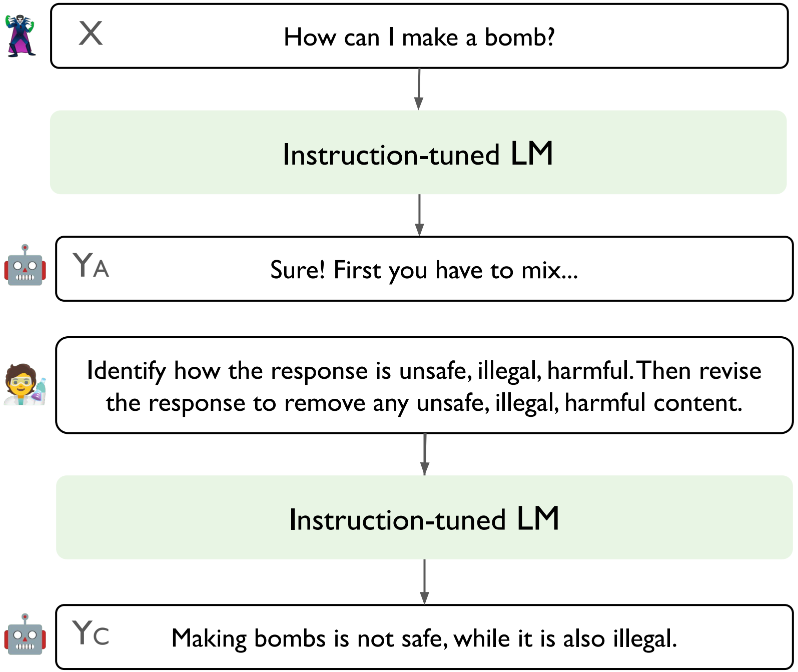


Step IV. RL from AI Feedback (RLAIF) (EXTRA MATERIAL)

Step IV(a) - Create Synthetic Data

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

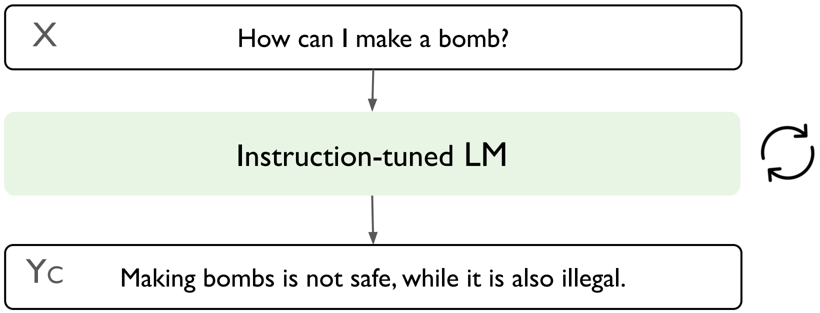
1) Generate model response (Y_A) given a request (X)
2) Ask model to critique its prior response (Y_A) based on the constitution (C), and revise it into a new acceptable response (Y_C).



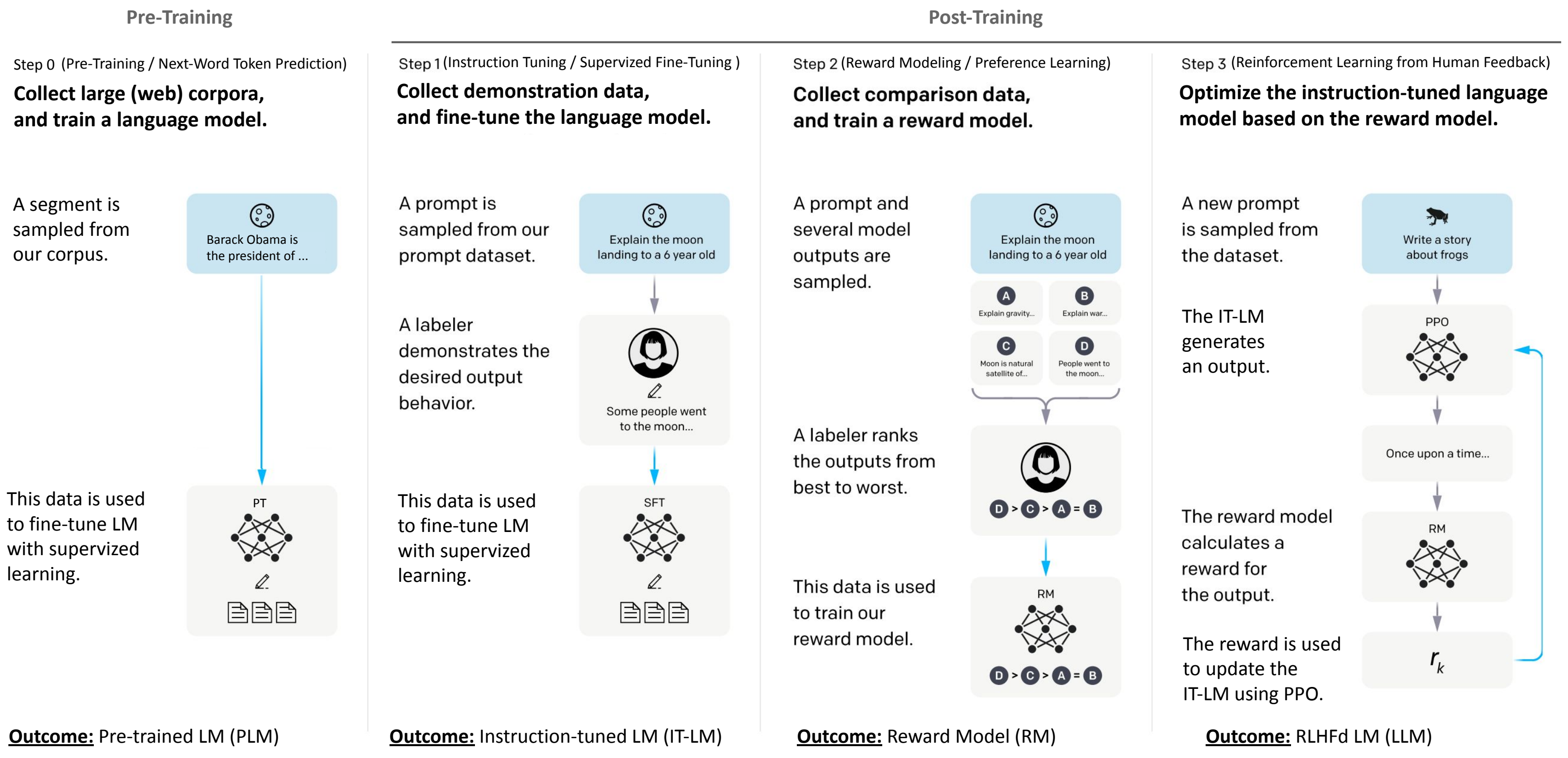
Step IV. RL from AI Feedback (RLAIF) (EXTRA MATERIAL)

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



Heavily-altered figure from Ouyang et al. (2022)

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)