

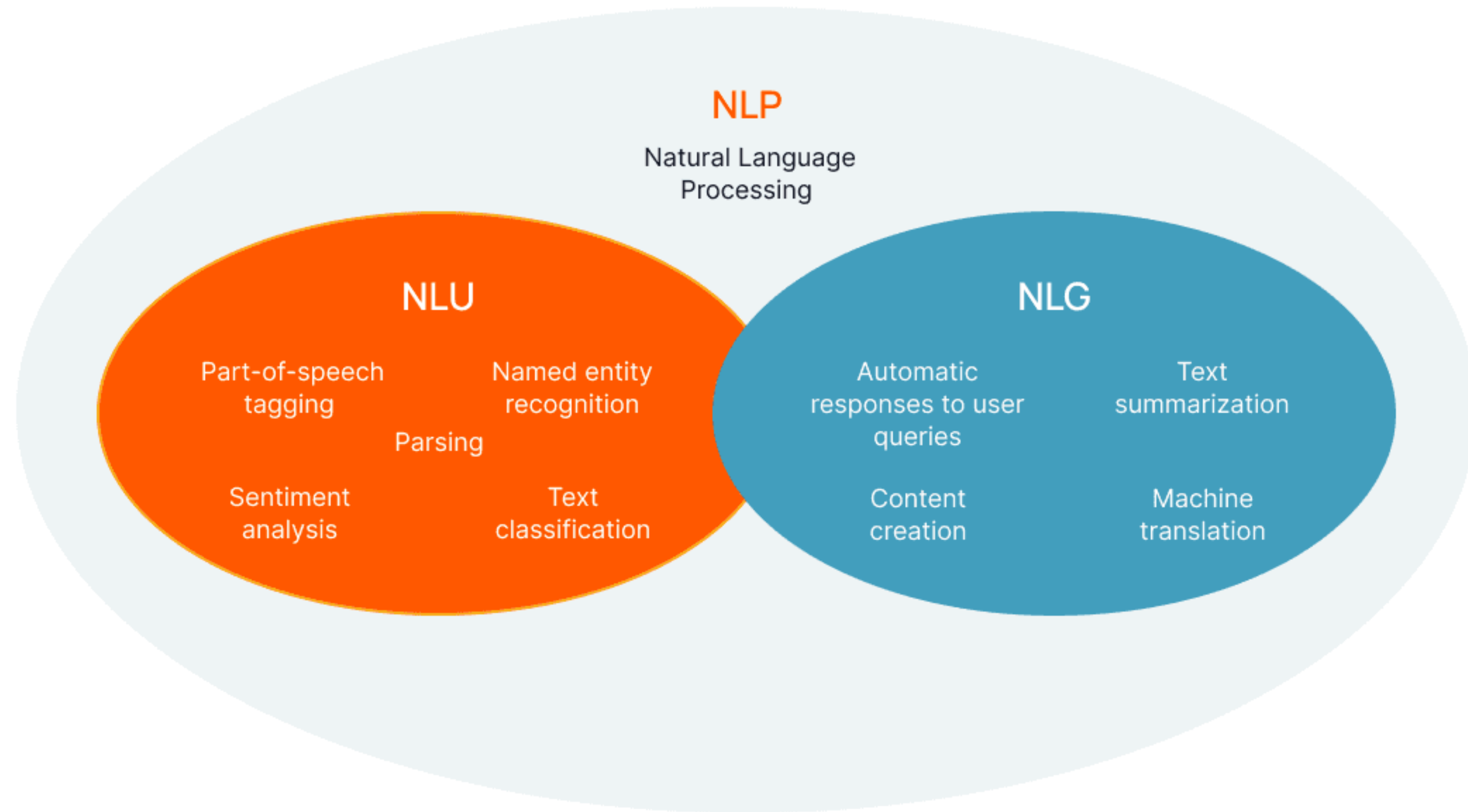
Fine-Tuning / Instruction Tuning

Haw-Shiuan Chang

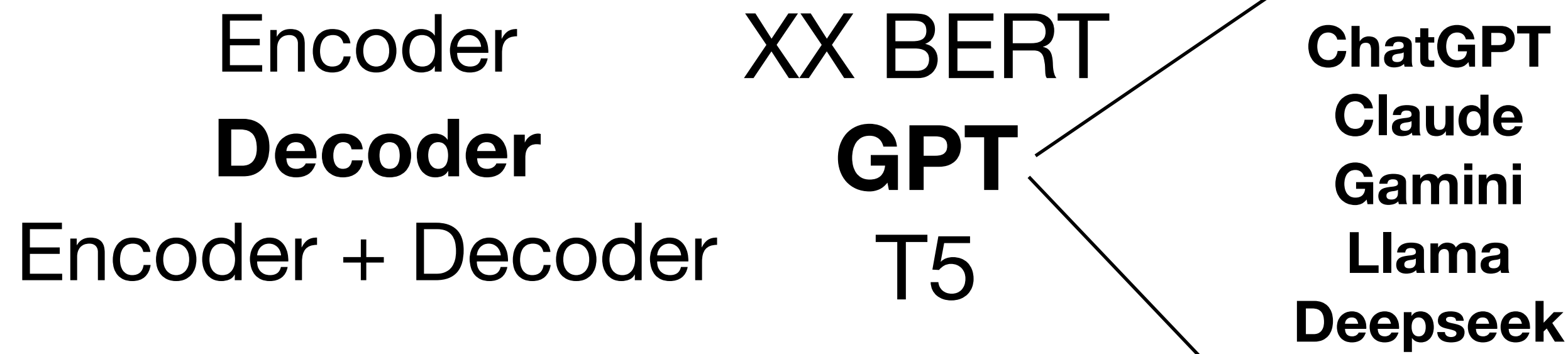
Deadlines

- <https://people.cs.umass.edu/~hschang/cs685/schedule.html>
- **3/3: Quiz 2 due**
- **3/7: Project proposals due**
 - Please submit only one proposal for each group
 - If the score of the proposal is lower than the final report, we will use the final report score.
 - However, if you don't submit the proposal, we won't give you the feedback and provide you with some LLM credit support.
 - **In your proposal, please estimate the cost of API credit you need and which LLM and service provider you plan to use.**
 - I know it is hard, but please try.
 - We only have \$500 for the whole class. Try not to have a project that needs hundreds of dollars (unless you are willing to pay by yourself).
 - You might get more money if your proposal looks better and more feasible
- **3/14: HW 1 due**
- **3/17: Quiz 3**

NLU vs NLG

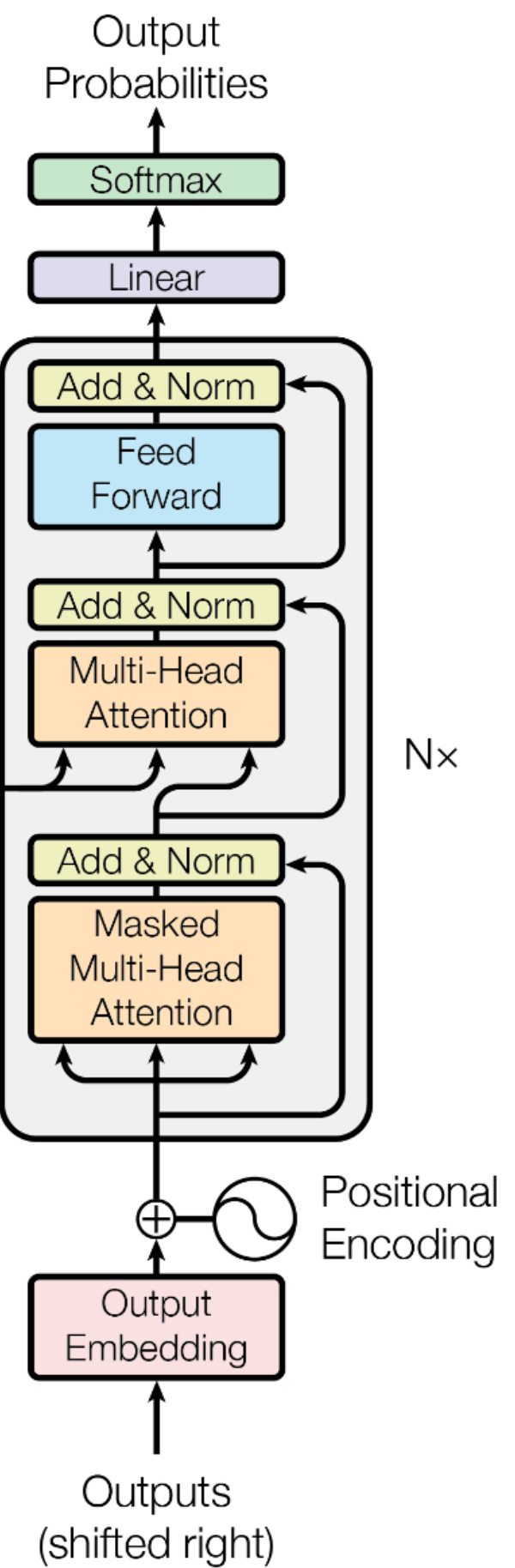


Architecture Comparison



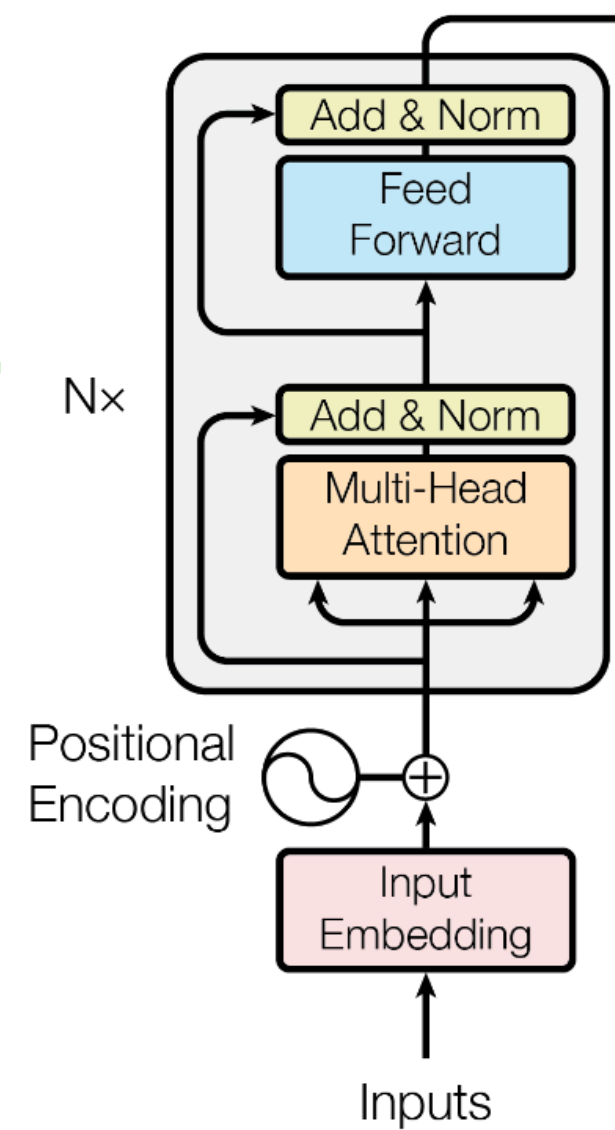
Encoder + Decoder

Loss?



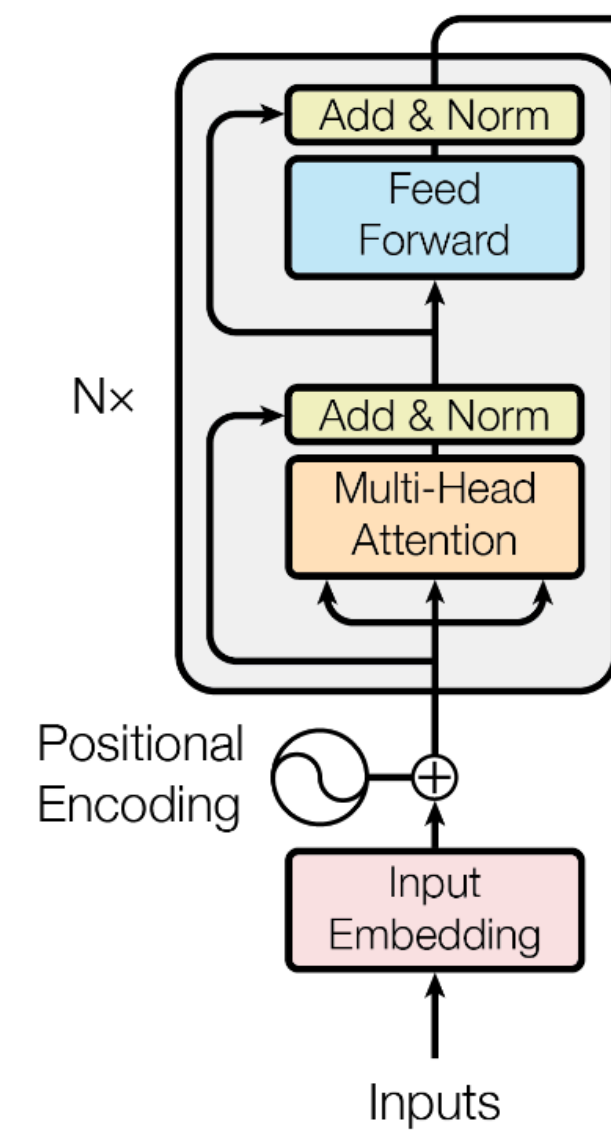
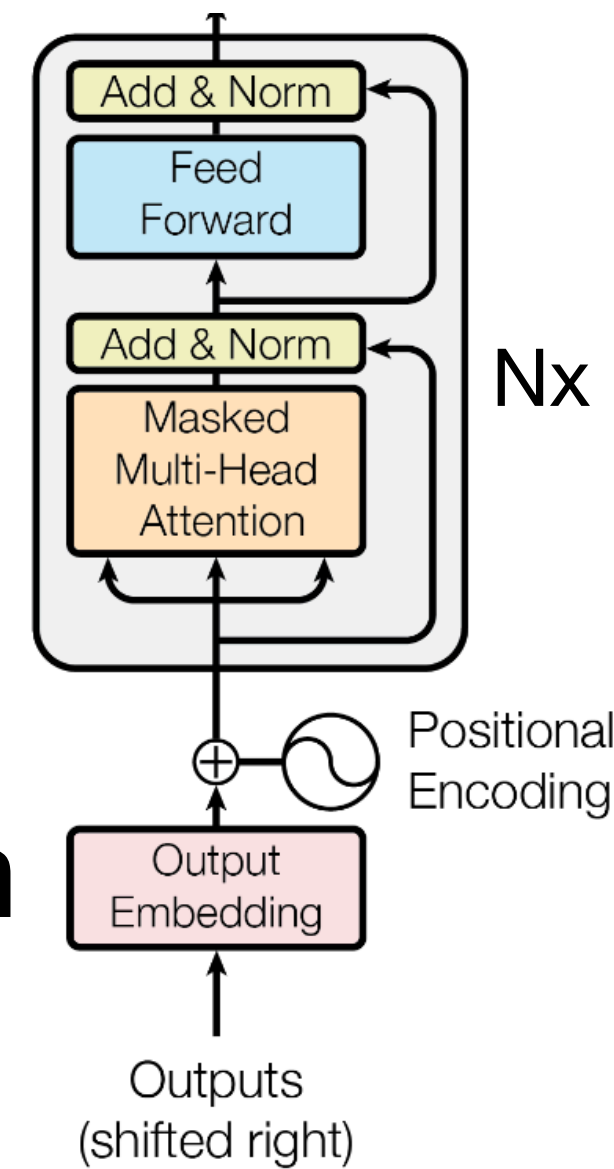
Encoder

Loss?



Decoder

Loss:
Predicting
next token



Positional Encoding

Inputs

Positional Encoding

Outputs
(shifted right)

Last Year Notes

LLM Development

- Architectures
 - MLP
 - RNN
 - Transformer
- Training Stages
 - Pretraining → Usually more expensive
 - **Supervised Fine-tuning (SFT)**
 - Alignment
 - Learning from Human Feedback (LHF)
 - Reasoning

Post-training stage

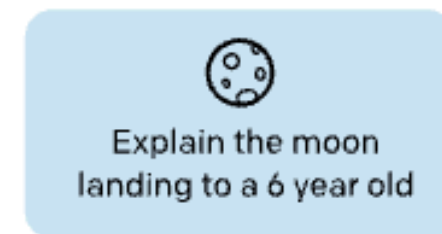


Supervised Fine-Tuning (SFT)

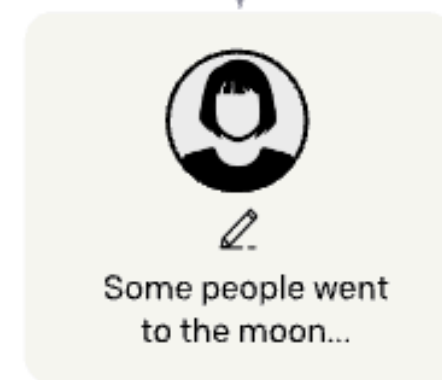
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



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

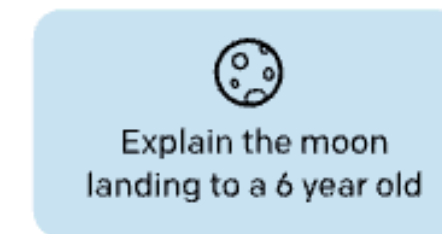


SFT

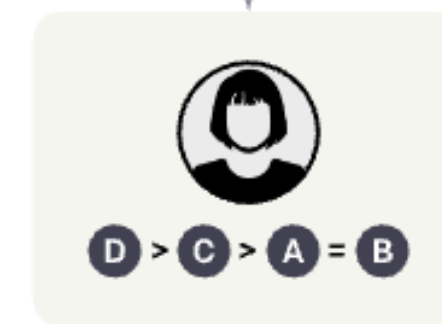
Step 2

Collect comparison data, and train a reward model.

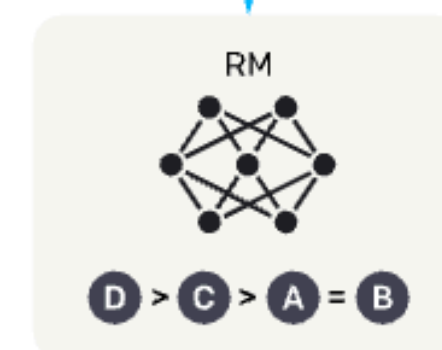
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

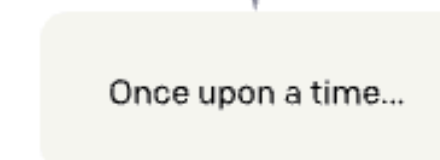
A new prompt is sampled from the dataset.



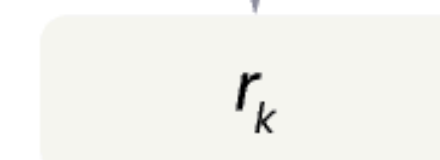
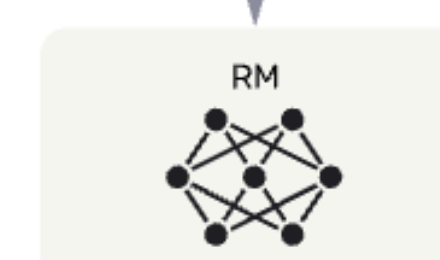
The policy generates an output.



The reward model calculates a reward for the output.



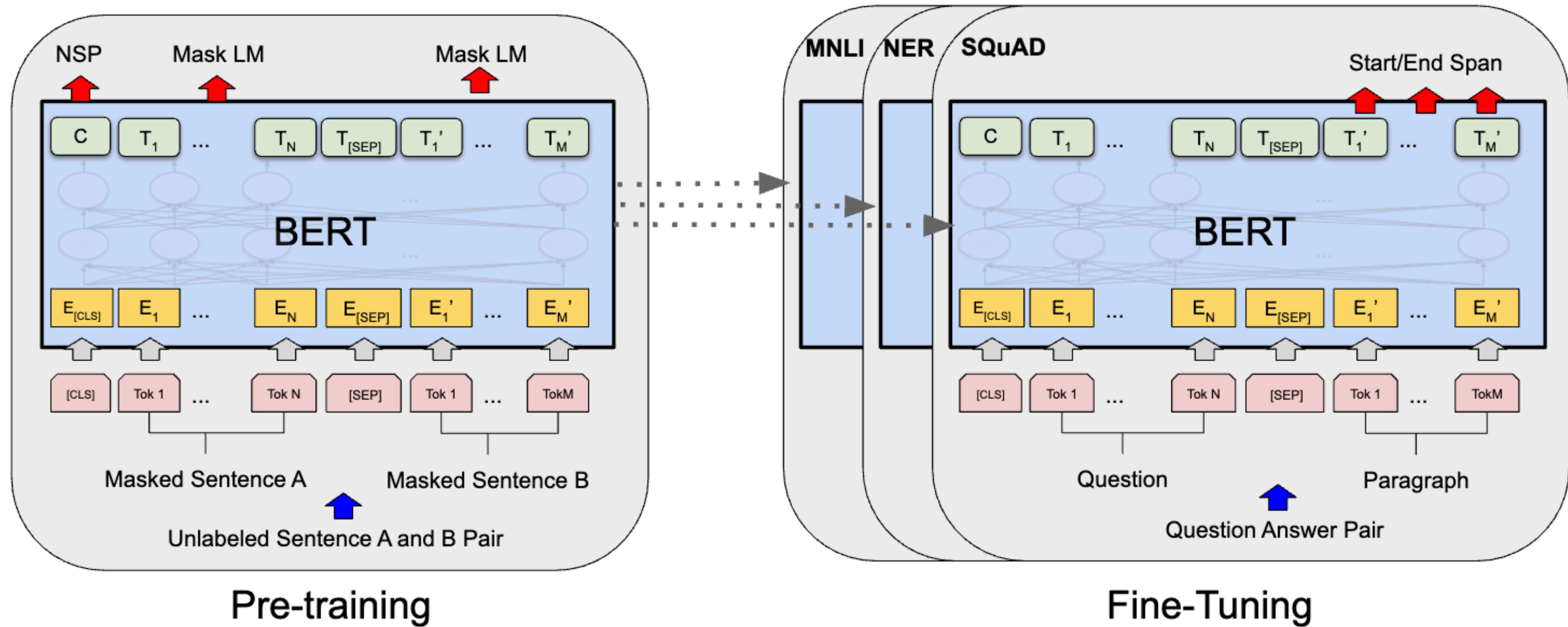
The reward is used to update the policy using PPO.



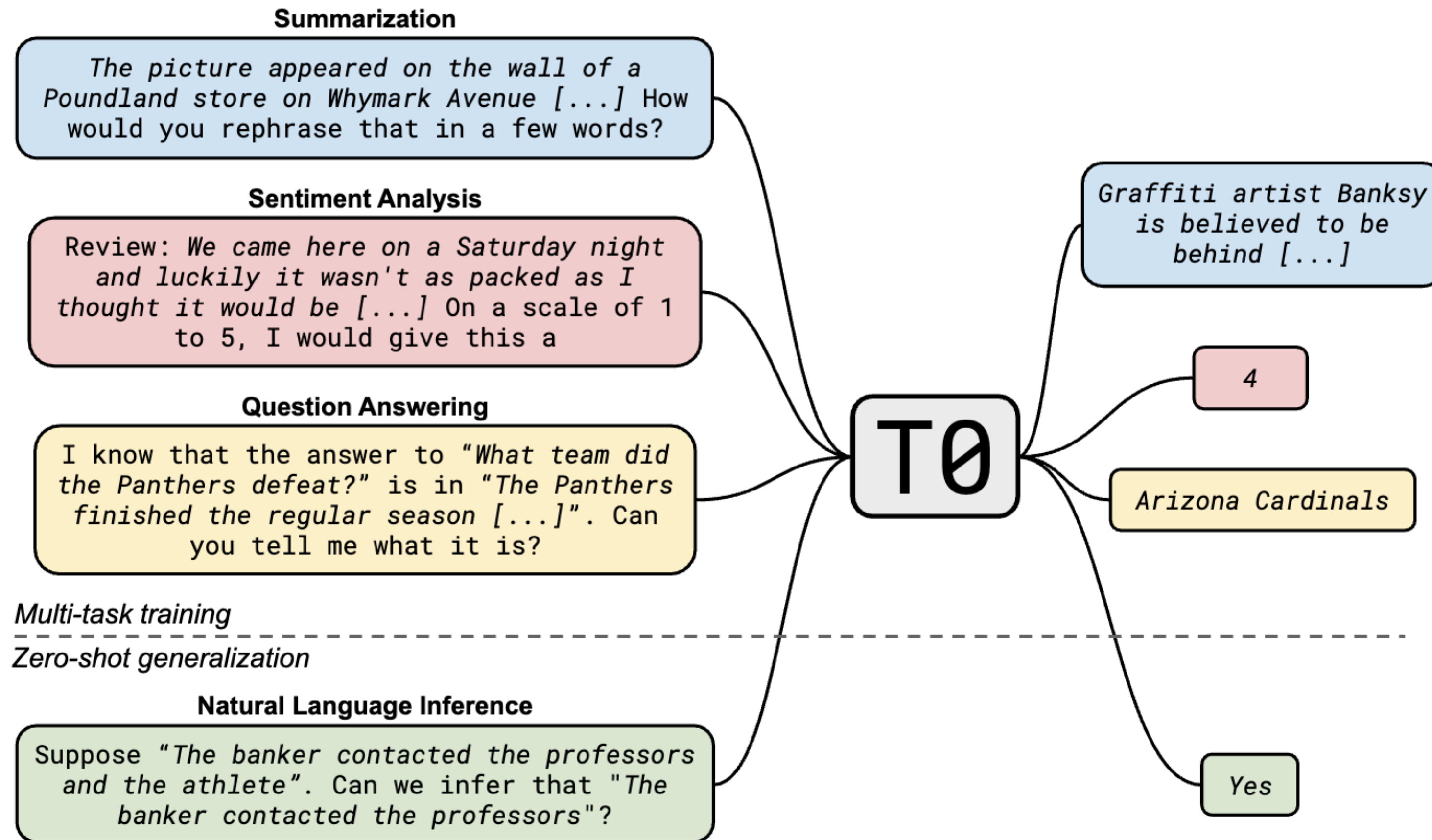
What Fine-Tuning Data Should we Collect/Use?

The answer is complicated

Old Fine-Tuning



Instruction Tuning



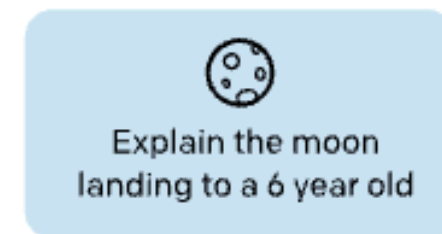
MULTITASK PROMPTED TRAINING ENABLES ZERO-SHOT TASK GENERALIZATION (<https://arxiv.org/pdf/2110.08207>)

Supervised Fine-Tuning (SFT)

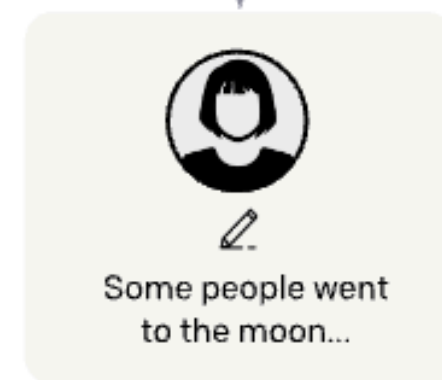
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



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

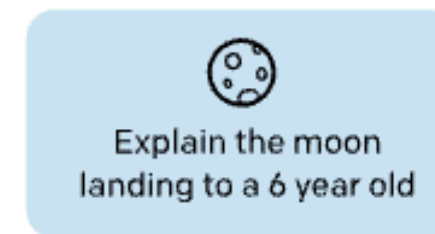


SFT

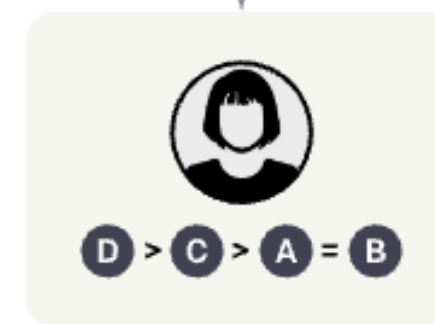
Step 2

Collect comparison data, and train a reward model.

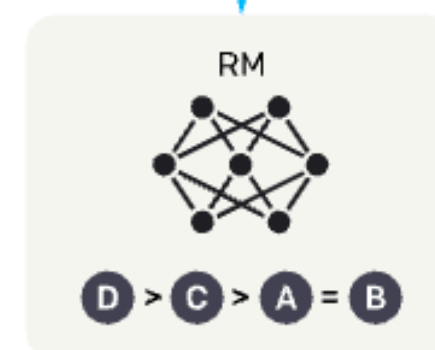
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

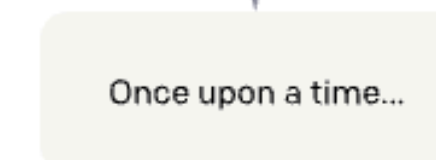
A new prompt is sampled from the dataset.



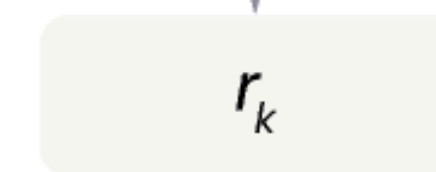
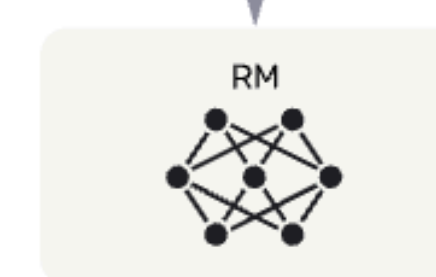
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Old Fine-Tuning

- Before ChatGPT/LLM
 - Models like BERT
- Tasks
 - Usually single Natural Language Understanding (NLU) task
- Input
 - No instruction
- Output
 - Often human labeled

Instruction Tuning

- Before ChatGPT/LLM
 - Models like T0
- Tasks
 - Many tasks, NLU + NLG
- Input
 - Instructions from hundreds of tasks
- Output
 - Often human labeled

SFT

- After ChatGPT/LLM
 - Models like GPT 3.5
- Tasks
 - Many tasks, usually Natural Language Generation (NLG) tasks
- Input
 - Mostly free-form Instructions
- Output
 - Sometimes extracted

Old Fine-Tuning

- Specialization
 - One FT Model for one task
- Cost
 - Cheap
- Boundary to Pretraining
 - Clear

Instruction Tuning

- Specialization
 - One FT Model for all tasks
- Cost
 - Expensive (various tasks)
- Boundary to Pretraining
 - Clear

SFT

- Specialization
 - One FT Model for all tasks
- Cost
 - Expensive (LLM)
- Boundary to Pretraining
 - Blurry

Difference between pretraining and fine-tuning is not loss different. It is data difference and hyperparameter difference.

The Blurry Boundary between Pretraining and SFT

Critics Reviews

[View All \(318\)](#)



Leonard Maltin

leonardmaltin.com

★ **TOP CRITIC**

The not-so-secret weapon this CAPTAIN AMERICA has going for it is Harrison Ford. Don't believe the nay-sayers out there: Brave New World is a 21st century Tall Tale, and if it takes two viewings to take it all in, so be it



Feb 21, 2025

[Full Review](#)



Lalo Ortega

Cine Premiere

...Captain America: A New World...is, quite simply, a tedious film on its own, and redundant in the grand scheme of things. [Full review in Spanish]



Rated: 1.5/5 • Feb 28, 2025

[Full Review](#)



Dino

★★★★★ **Do your taxes the easy way.**

Reviewed in the United States on February 7, 2025

Platform For Display: PC/MAC Download | Edition: Deluxe - Federal & State | **Verified Purchase**

For years I used an accounting firm to do my taxes, and they always sent me a multipage form that asked a series of questions and space to answer them. Turbo Tax asks the same questions, and when you answer them, you're filling out your tax forms. The accounting firm charged a base fee plus by the tax form. It was expensive. Save money by using Turbo Tax.

6 people found this helpful

Helpful

Report



Judy L.

★★★★☆ **Pretty straight forward, but some areas can be difficult to figure out the correct entries.**

Reviewed in the United States on February 4, 2025

Platform For Display: PC/MAC Download | Edition: Deluxe - Federal & State | **Verified Purchase**

I have used Turbo Tax software for quite a few years to do our taxes. We are at a point that we don't have much in deductions and have used the standard deduction for the last few years. Even so, I feel more comfortable with the option to use deductions if we can save more than using the standard deduction. Since tax laws are constantly changing, I still run through all of the possible deductions which are, for the most part, described well in this software. I suppose I could use the online version to save the cost of this software, but I don't feel comfortable putting all of our financial information online.

Very happy to see the search agent Search-R1 and our RAGEN codebase has been able to support it 😊

We put a lot of efforts to make the RAGEN codebase easy to reuse/follow.

Welcome to play with RAGEN for agent framework using simple RL recipe like DeepSeek R1

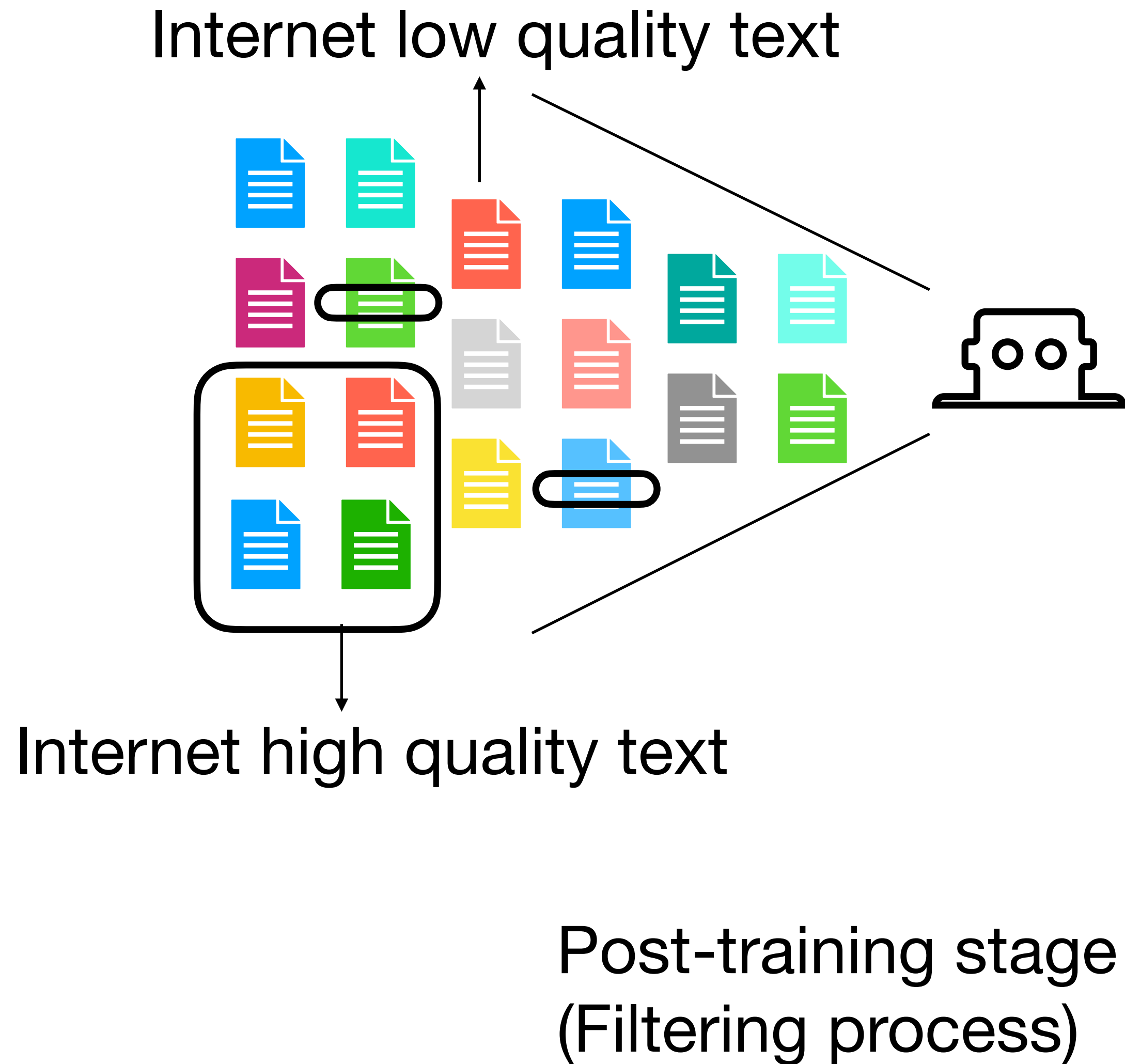
- “Transfer” to Sentiment Analysis

- Controlled Review Generation

Perspectives \neq Facts

I will present my conclusions derived from existing papers and my experience.
Some of them have not been universally accepted by the NLP community yet

LLM Development



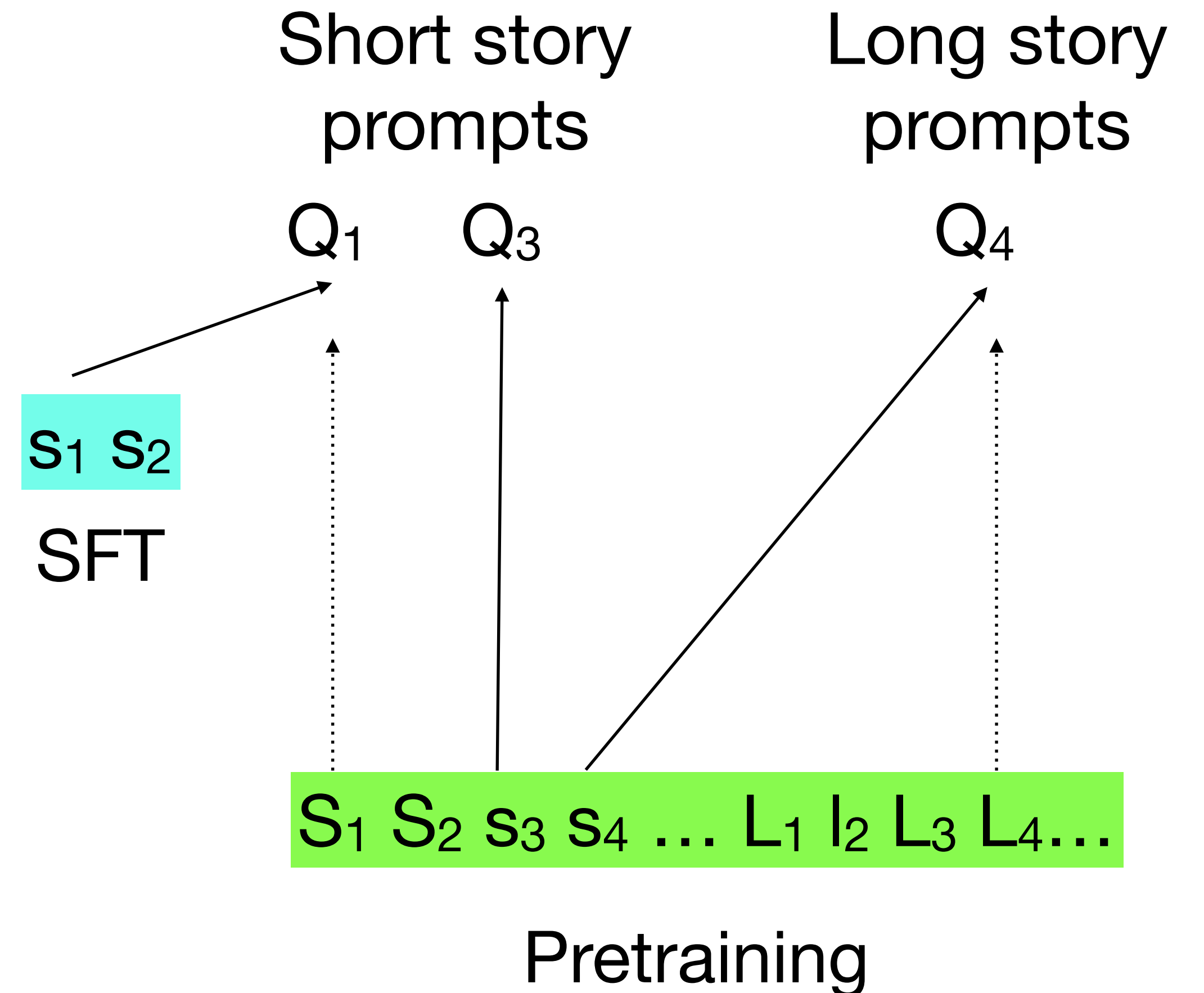
- Architectures
 - MLP
 - RNN
 - Transformer
- Training Stages
 - Pretraining
 - **Supervised Fine-tuning (SFT)**
 - Alignment
 - Learning from Human Feedback (LHF)
 - Reasoning

Question

- Assuming you are in an SFT team at a large company. You recently collected 1k high-quality (constraints, short stories) to improve an LLM.
- Given the context is “Please output a short story with the following constraints: {constraints}.”, you fine-tune the LLM to output the collected short story.
- During the testing time, you compare the LLM’s response before and after your fine-tuning given the prompt “Please output a long story.”
- After fine-tuning, will the story length be reduced a lot?

Fine-tuning LLM for a Target Task

- Target task itself
 - Response from the fine-tuning data
 - Could also overwrite the good responses in the pretraining data
 - Response learnt from pretraining
 - Inducing output that is similar to the output of the fine-tuning data
- Other tasks
 - Could make the output closer to the output of the target task
 - e.g., after training on a short story, the story length would decrease when you want LLM to give you a long story



LLM Paradigm Shift

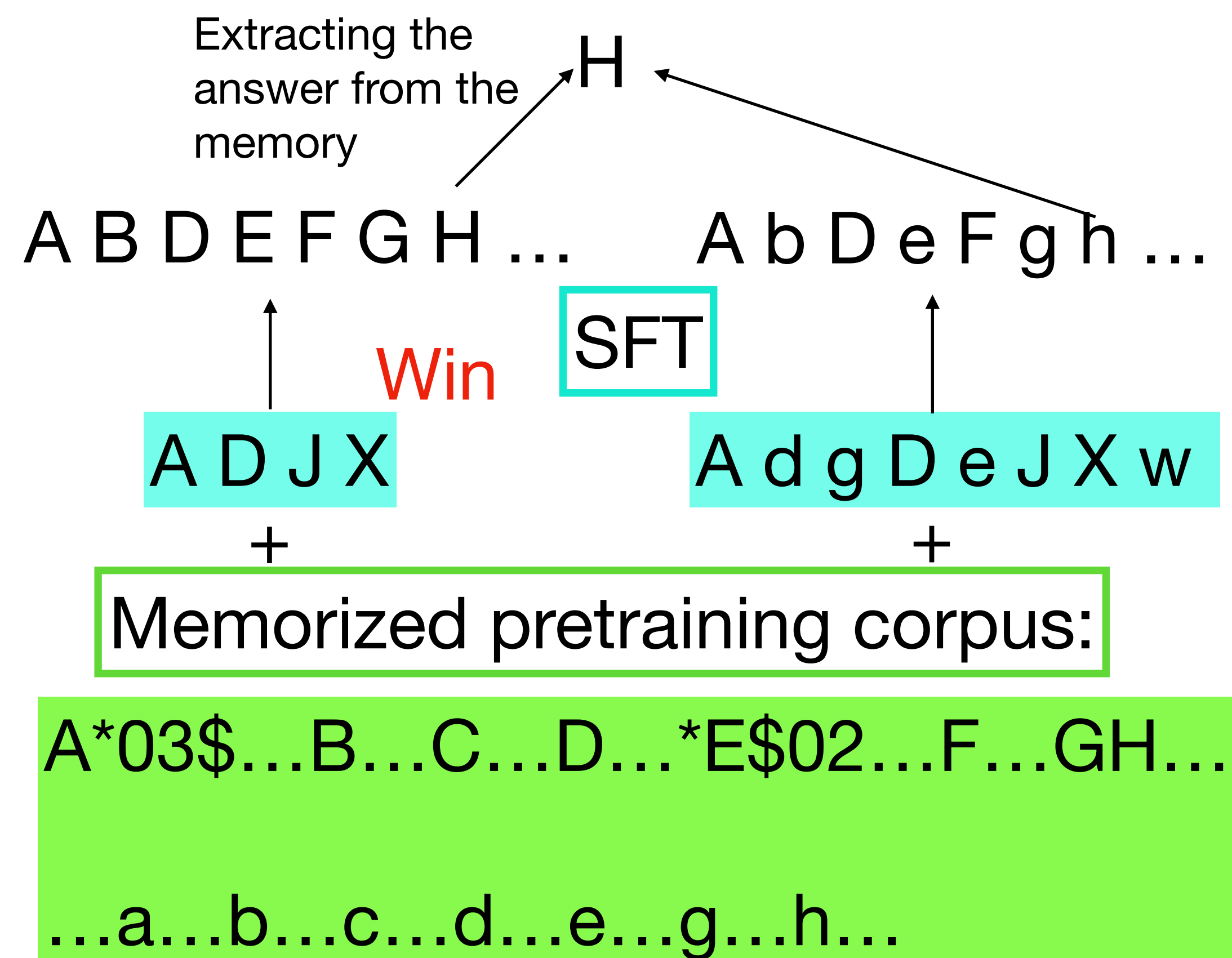
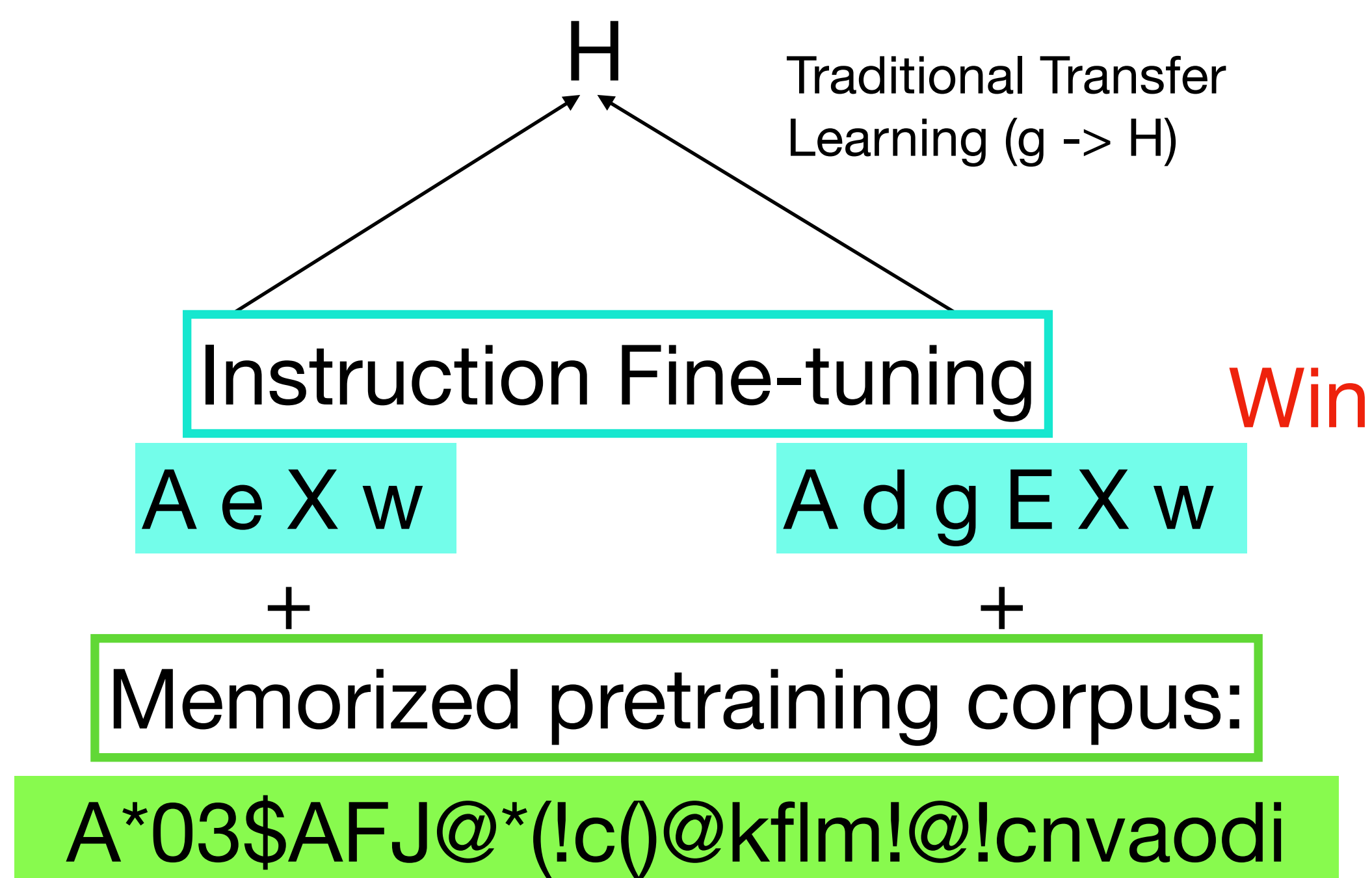
- Before LLM, instruction tuning should also use as many tasks/data in the fine-tuning stage as possible
 - Example: Flan-T5:
 - Smaller encoder-decoder model, less pretraining -> many fine-tuning tasks
- After LLM, SFT should only use a few high-quality fine-tuning data
 - Example ChatGPT:
 - Larger decoder-only model, more pretraining -> fewer fine-tuning tasks

Question

- Remember that the loss function for SFT/pretraining/instruction-tuning is the same.
- A larger pretraining dataset is better.
- A larger instruction-tuning dataset is better.
- Then, why could a larger SFT dataset degrade the performance?

Why Could Fewer Data be Better?

- First task -> A: high-quality data, a: low-quality data



Recent studies show that such transfer learning does not actually work generally. See this paper:

Do Models Really Learn to Follow Instructions? An Empirical Study of Instruction Tuning (<https://arxiv.org/pdf/2305.11383>)