Instruction Tuning and Reinforcement Learning from Human Feedback

IE686 Large Language Models and Agents

Credits

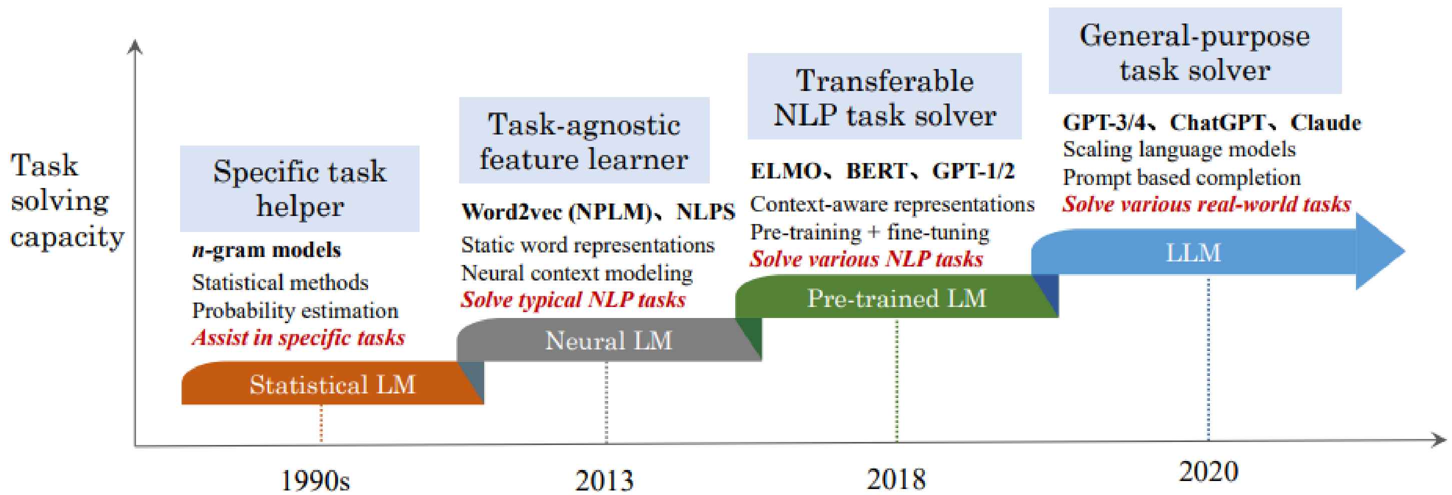
- This slide set is based on slides from
  - **Jiaxin Huang**
  - Mrinmaya Sachan
  - Tatsunori Hashimoto

- Many thanks to all of you!

Outline

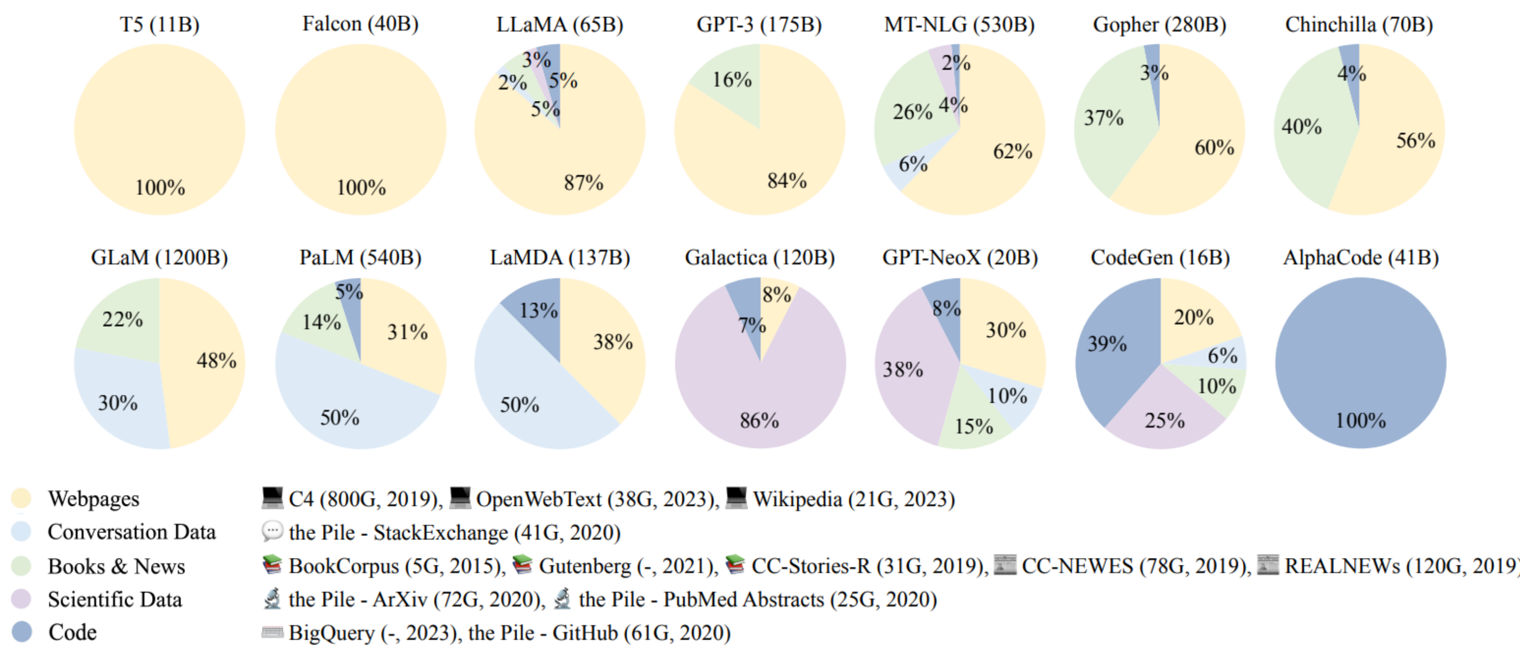
- **Recap: Pre-training Language Models**
- Scaling up and Emergent Abilities of LLMs
- Instruction Tuning
- Reinforcement Learning from Human Feedback
- Existing Large Language Models

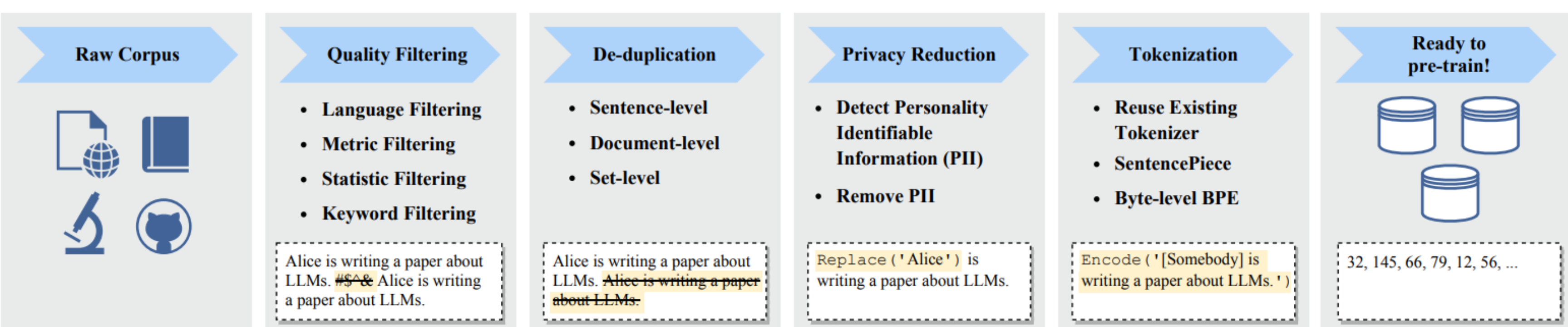
Recap: Language Models over Time



- Simple n-gram models followed by shallow neural methods and RNNs
- The Transformer architecture started the age of pre-trained language models
  - Large-scale Pre-training followed by task-specific fine-tuning
    - ➔ Transfer Learning

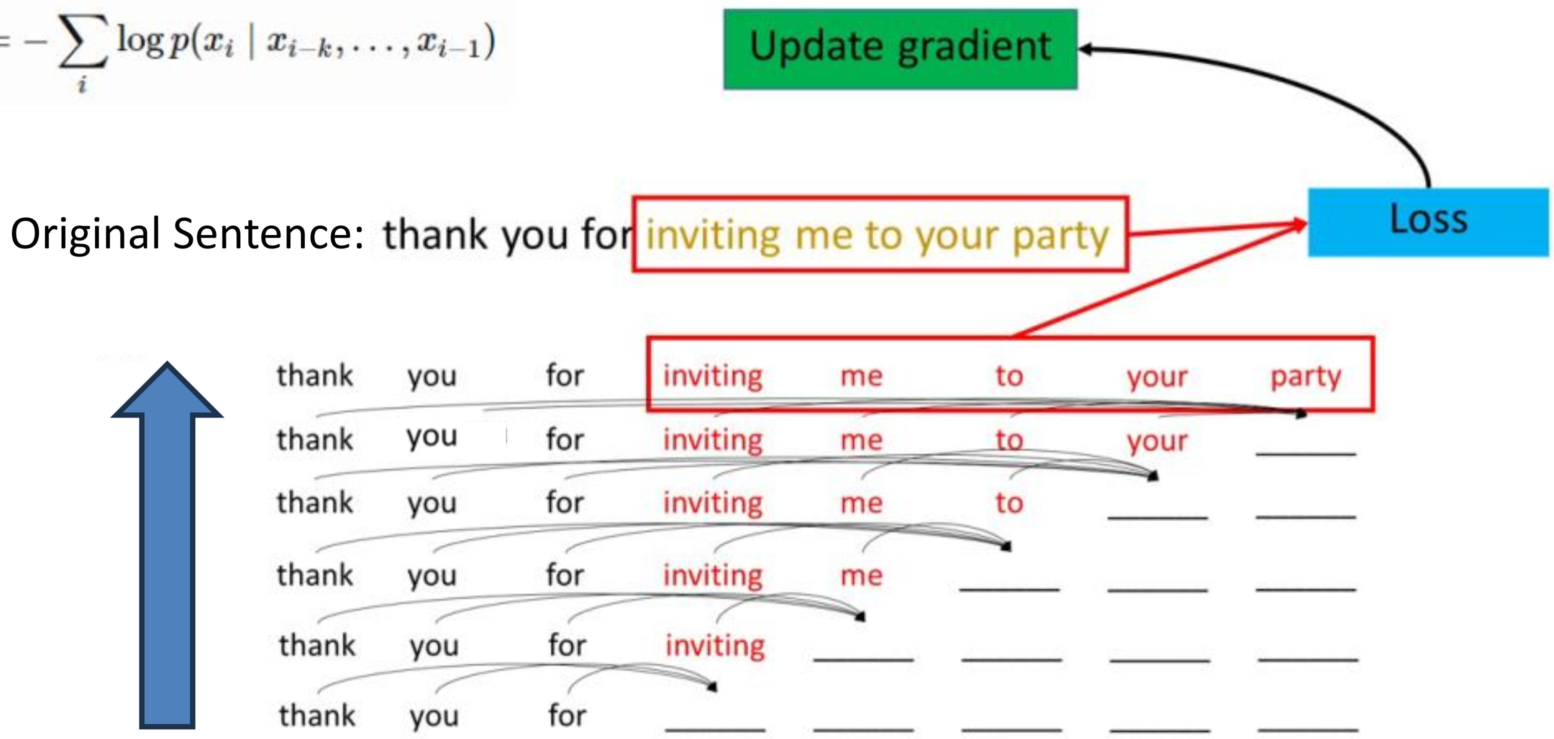
Recap: Pre-training Data





Recap: Pre-training Decoder-only

$$\mathcal{L}_{\text{LM}} = -\sum_i \log p(x_i \mid x_{i-k}, \ldots, x_{i-1})$$



Language Modeling ≠ Solving Tasks

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION 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.
```

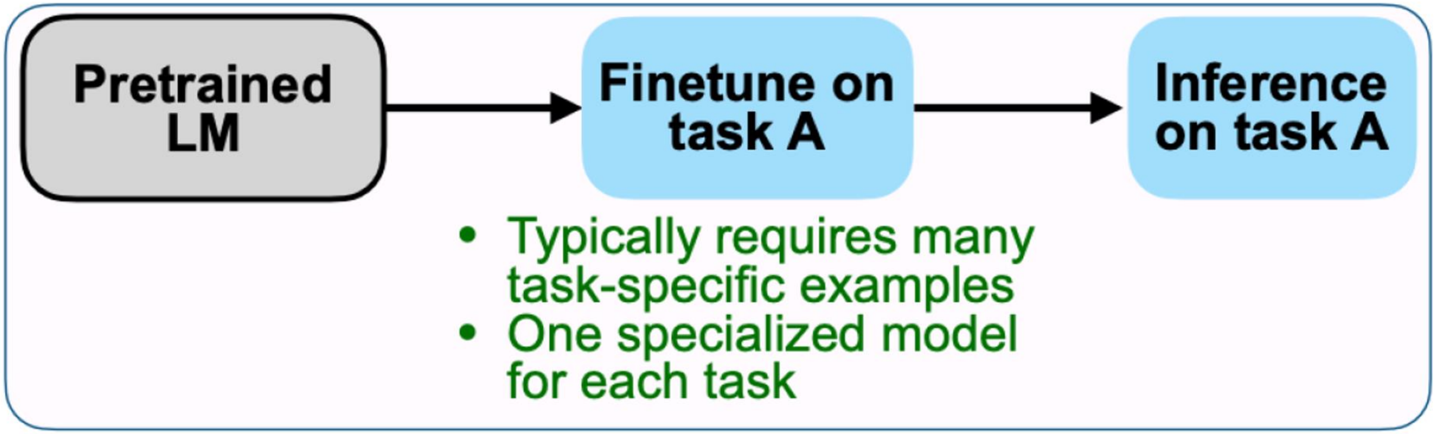
- Language modelling with **next token prediction** does not make the model a competent task solver
- How to adapt to correctly solving tasks?

Ouyang, L et al., 2022. Training Language Models to follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, *35*, pp.27730-27744.

Pre-train/Fine-tune Paradigm of PLMs

- The pre-training stage lets language models learn generic representations and knowledge from **large** corpora, but they are not fine-tuned on any form of user tasks.
- To adapt language models to a specific downstream task, use **comparably small** task-specific datasets for fine-tuning

➔ Transfer knowledge from pre-training, show the model what we want the output to look like and subsequently perform well on **one** task

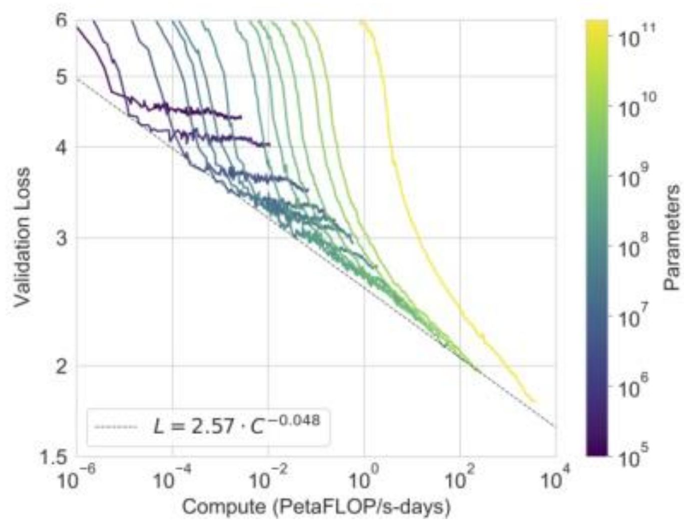


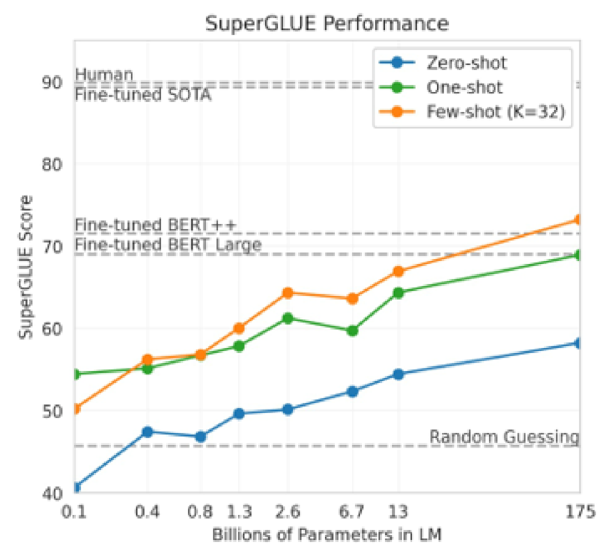
Outline

- Recap: Pre-training Language Models
- **Scaling up and Emergent Abilities of LLMs**
- Instruction Tuning
- Reinforcement Learning from Human Feedback
- Existing Large Language Models

Scaling up Language Models

- Scaling in three dimensions has been shown to strongly increase task solving capability and generalization
  - **Model size** in terms of parameters
  - Increasing pre-training **data**
  - Available training **compute**





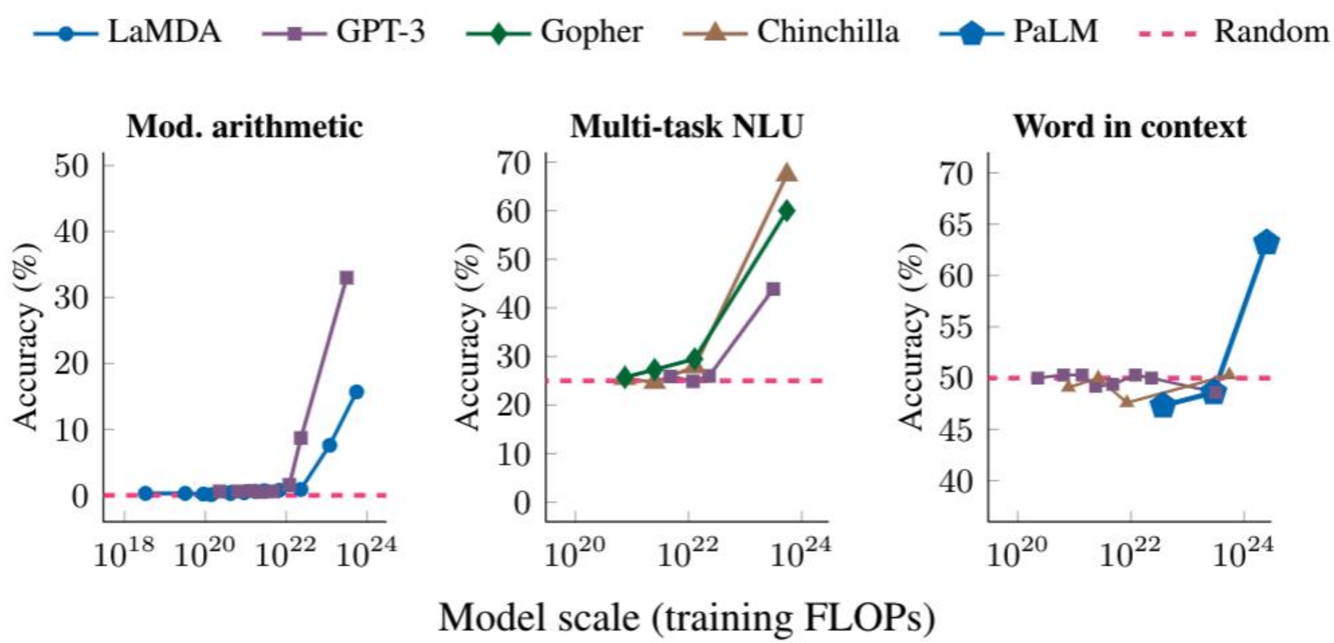
Emergent Abilities of LLMs

- "Abilities that are not present in small models but arise in large models"

  J. Wei et al., "Emergent Abilities of Large Language Models," CoRR, vol. abs/2206.07682, 2022

- Three typical emergent abilities:
  - **In-context learning**: After providing the LLM with one or several task demonstrations in the prompt, it can generate the expected output (next week)
  - **Instruction following**: Fine-tuning the model with instructions for various tasks at once, leads to strong performance on unseen tasks (instruction tuning -> our focus today)
  - **Step-by-step reasoning**: LLMs can perform complex tasks by breaking down a problem into smaller steps. The chain-of-thought prompting mechanism is a popular example (next week)

Emergent Abilities of LLMs

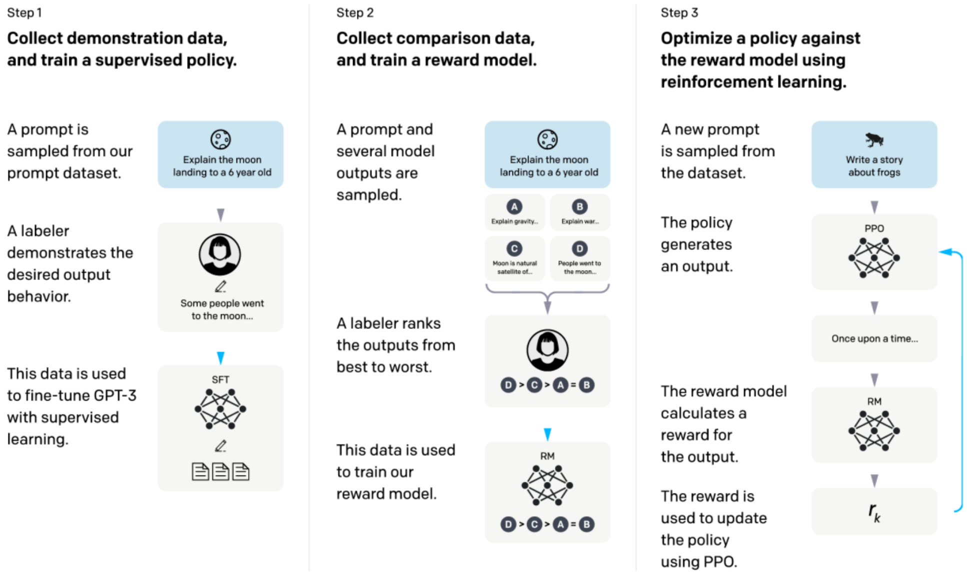


- Emergent abilities can lead to sudden leaps in performance on various tasks

J. Wei et al., "Emergent Abilities of Large Language Models," CoRR, vol. abs/2206.07682, 2022

Typical LLM Training Procedure

1. Self-supervised pre-training (**next token prediction**)
2. Supervised training on pairs of human-written prompt/answer pairs (Step 1)
3. LLM tasked to generate multiple outputs for a prompt, which are ranked by a human and used to train a reward model (Step 2)
4. The LLM is optimized with reinforcement learning using the reward model (Step 3)

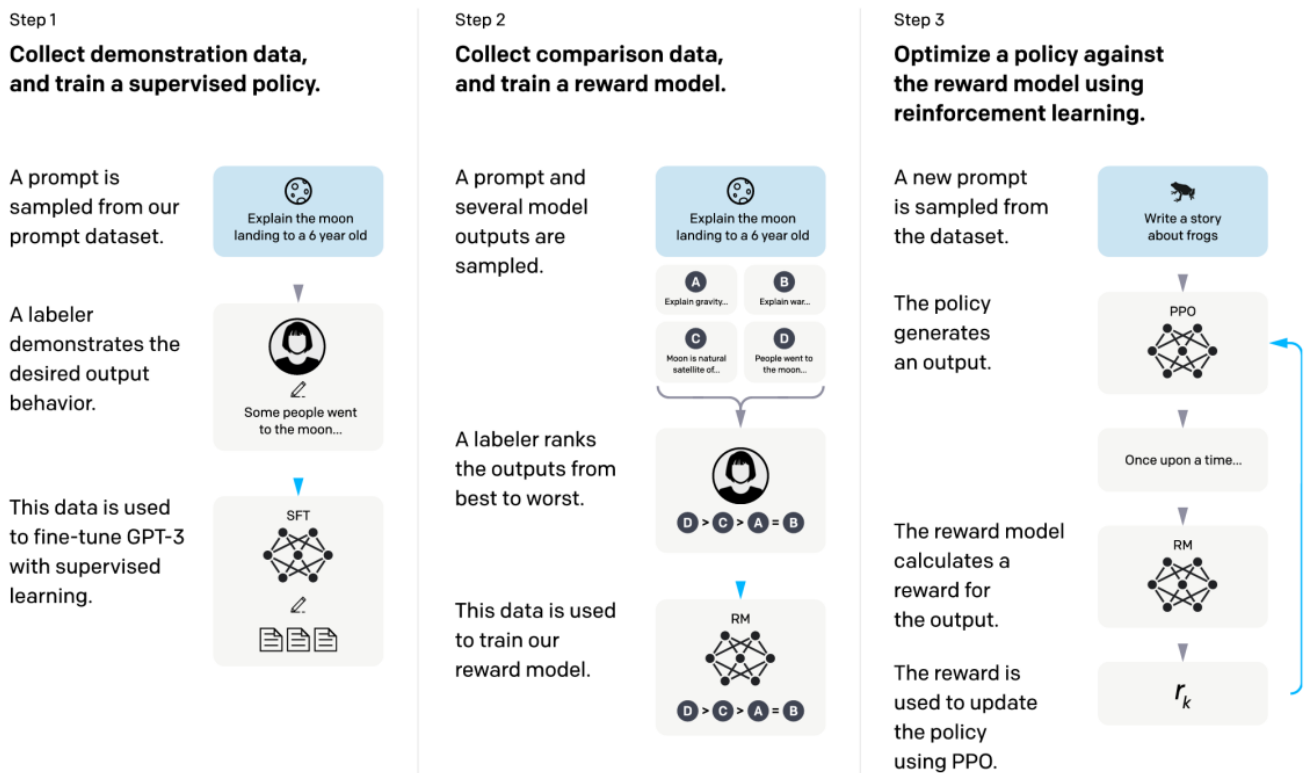


Ouyang, L et al., 2022. Training Language Models to follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, *35*, pp.27730-27744.

Outline

- Recap: Pre-training Language Models
- Scaling up and Emergent Abilities of LLMs
- **Instruction Tuning**
- Reinforcement Learning from Human Feedback
- Existing Large Language Models

LLM Training Framework



Instruction-Tuning

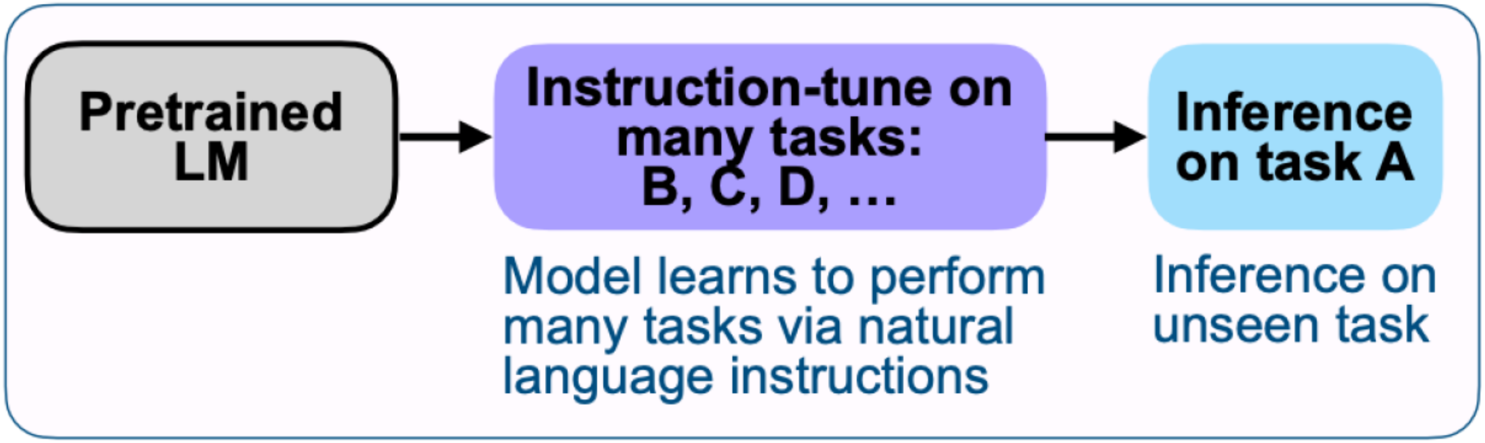
Reinforcement Learning from Human Feedback

Instruction Tuning

- Leverage emergent ability of the models
- Incorporate instructions into the fine-tuning procedure by prepending a "description" of each task to be carried out
- Examples
  - Sentiment -> "Is the sentiment of this movie review positive or negative?"
  - Translation (En to De) -> "Translate the following sentence into German:"
  - …
- Some simple templates are used to transform existing datasets into an instructional format

Instruction Tuning

- Fine-tune on many tasks at once
- Teaches language model to follow different natural language instructions, so that it can perform well on downstream tasks and even **generalize** to unseen tasks



Increasing Generalization

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Generalization	Task	Dataset	Category	Instances
new examples	=	=	=	≠
new category	=	=	≠	≠
new domain	=	≠	≠	≠
new skill	≠	≠	≠	≠

Increasing generalization

Instruction Tuning: Adding Diversity

- There is a gap between NLP tasks and user needs…

In traditional NLP, "tasks" were defined as subproblem frequently used in products:
- Sentiment classification
- Text summarization
- Question answering
- Machine translation
- Textual entailment

Narrow definitions of tasks.
Not quite what humans want, nevertheless, it might be a good enough proxy.
Plus, we have lots of data for them.

What humans need:
- "Is this review positive or negative?"
- "What are the weaknesses in my argument?"
- "Revise this email so that it's more polite."
- "Expand this this sentence."
- "Eli5 the Laplace transform."
- …

Quite diverse and fluid.
Hard to fully define/characterize.
We don't fully know them since they just happen in some random contexts.

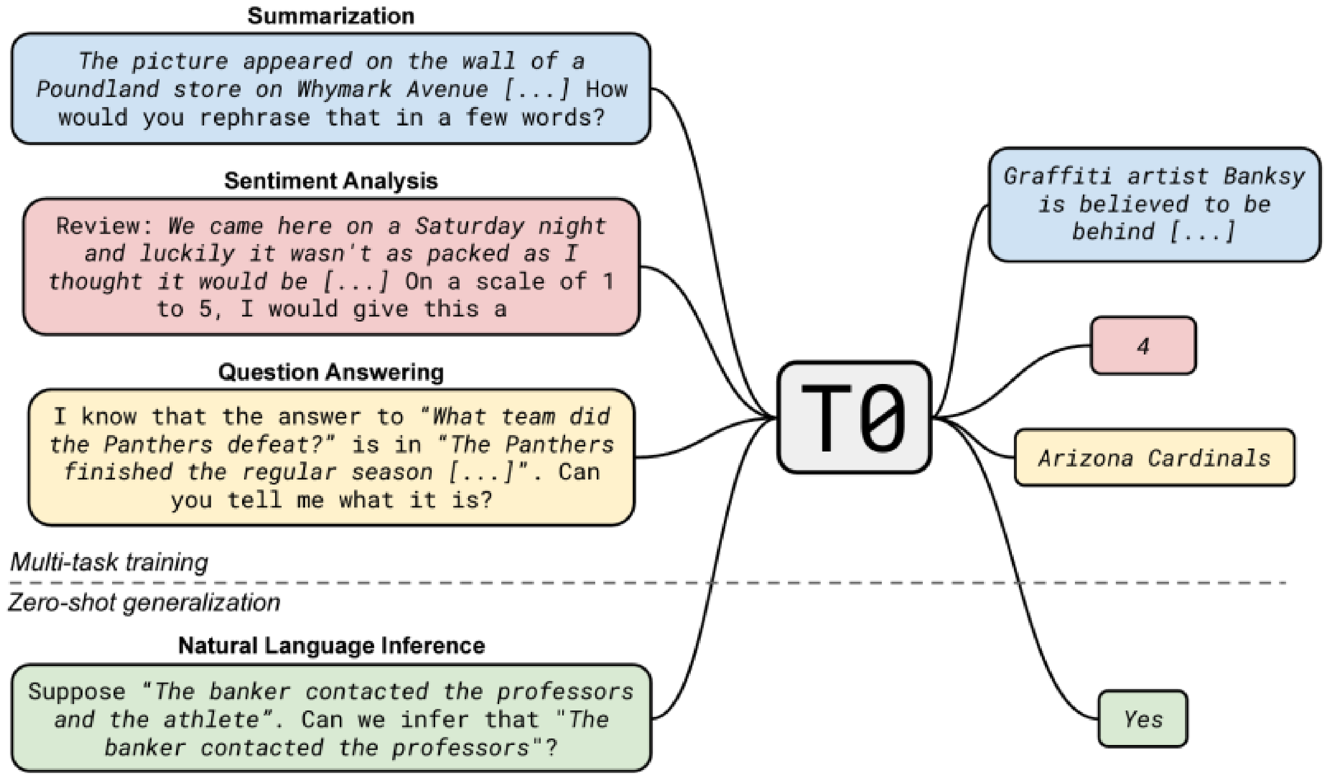
- More diversity needs to be added to the data...

Adding Diversity via Task Prompts

- Example Task: Summarization
- Create diversity from the **same example** via prompt variations

```
"Write highlights for this article:\n\n{text}\n\nHighlights: {highlights}"
"Write a summary for the following  article:\n\n{text}\n\nSummary: {highlights}"
"{text}\n\nWrite highlights for this article. {highlights}"
"{text}\n\nWhat are highlight points for this article? {highlights}"
"{text}\nSummarize the highlights of this article. {highlights}"
"{text}\nWhat are the important parts of this article? {highlights}"
"{text}\nHere is a summary of the highlights for this article: {highlights}"
"Write an article using the following points:\n\n{highlights}\n\nArticle: {text}"
"Use the following highlights to write an article:\n\n{highlights}\n\nArticle:{text}"
"{highlights}\n\nWrite an article based on these highlights. {text}"
```

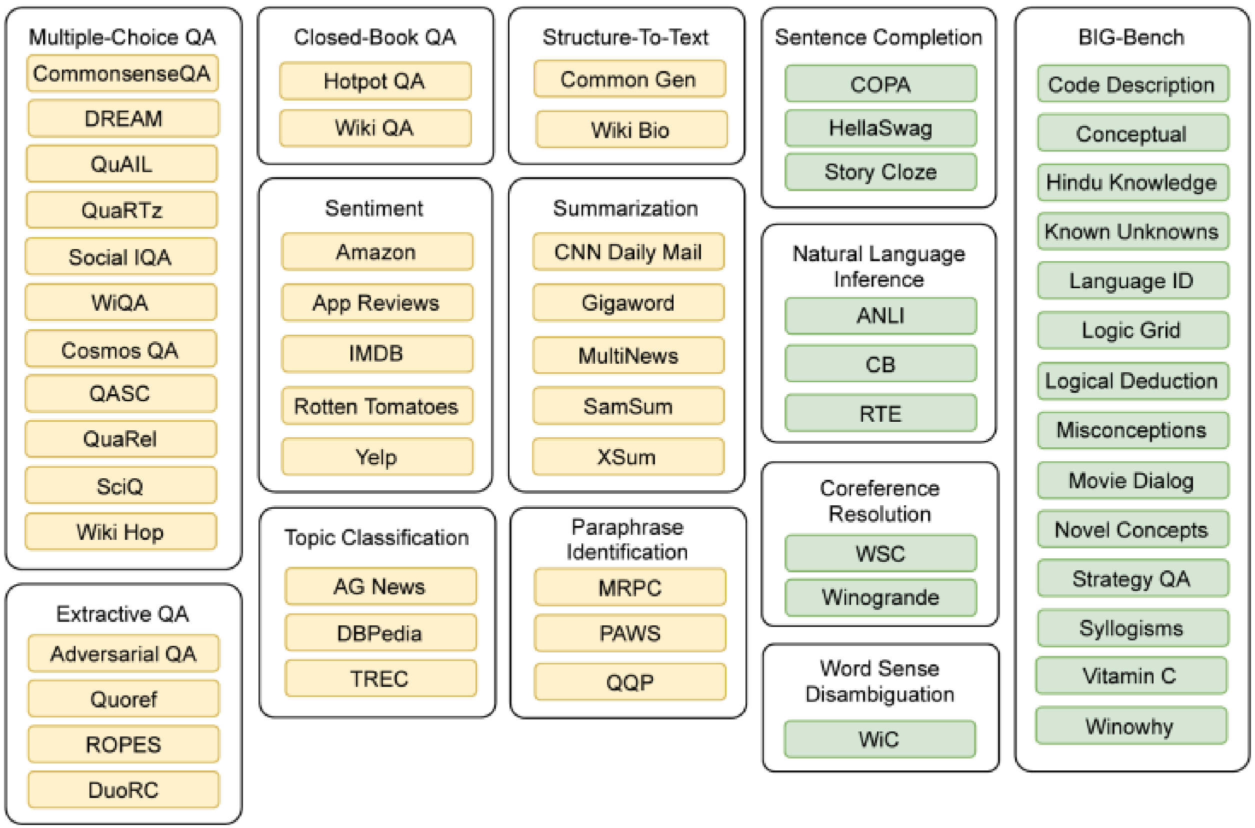
T0 – An Instruction-tuned LLM



Sanh, V. et al., Multitask Prompted Training Enables Zero-Shot Task Generalization.
In *International Conference on Learning Representations*.

T0 Training Sets

- Collected from multiple public NLP datasets and variety of tasks

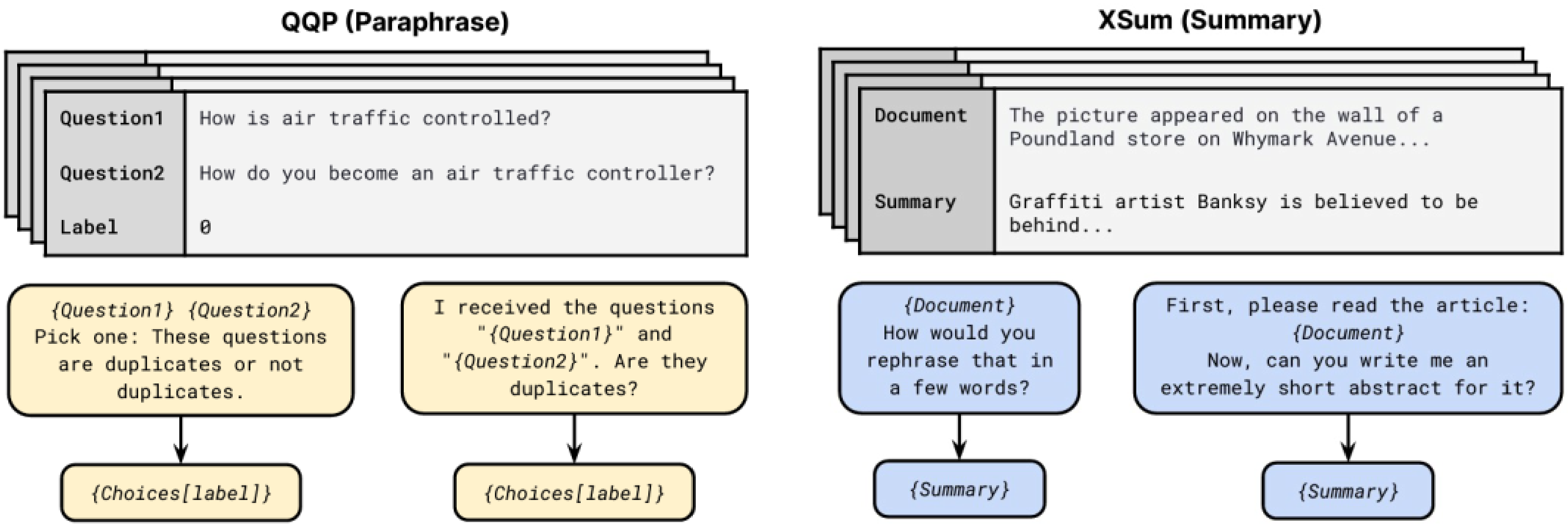


Training Mixtures and Unseen Sets

- Training Mixtures:
  - Question answering, structure-to-text, summarization
  - Sentiment analysis, topic classification, paraphrase identification
- Unseen test set:
  - Sentence completion, BIG-Bench
  - Natural language inference, coreference resolution, word sense disambiguation
- T0 is trained using the T5 transformer (11B model)

Task Adaptation with Prompt Templates

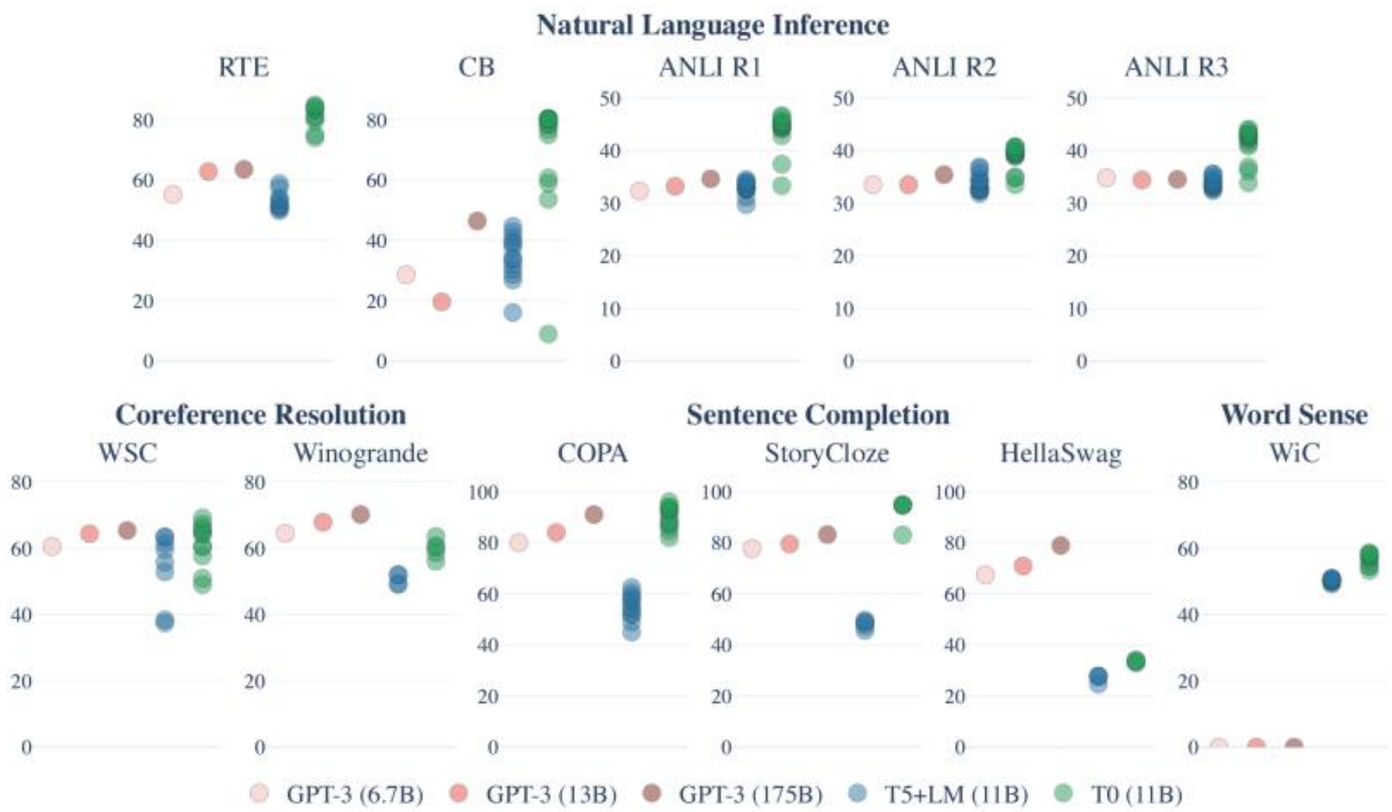
- Instead of directly using input/output pairs, specific instructions are added to explain each task
- The outputs are natural language tokens instead of class labels



Performance on Unseen Tasks

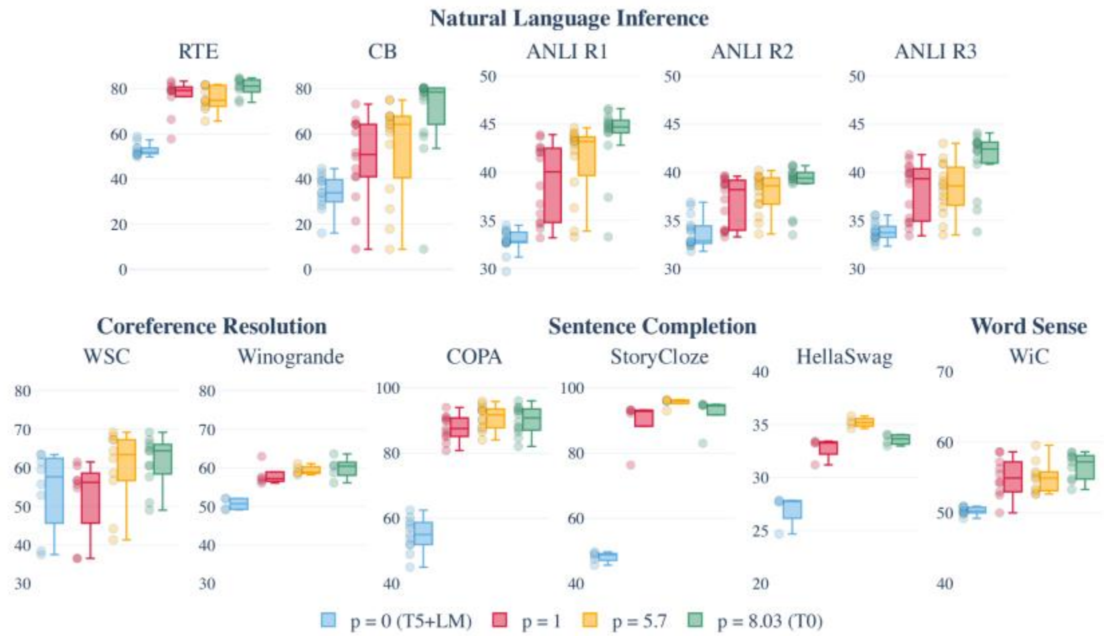
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- For T5 and T0, each dot represents one evaluation prompt



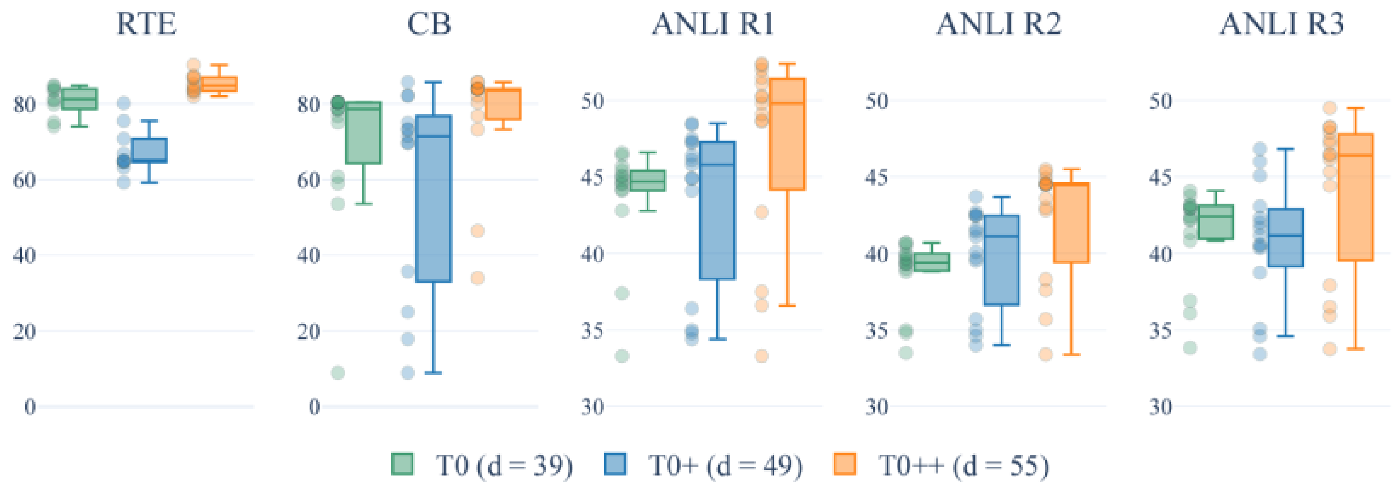
Effect of Prompt Variations

- Increasing the number of paraphrasing prompts generally leads to better performance



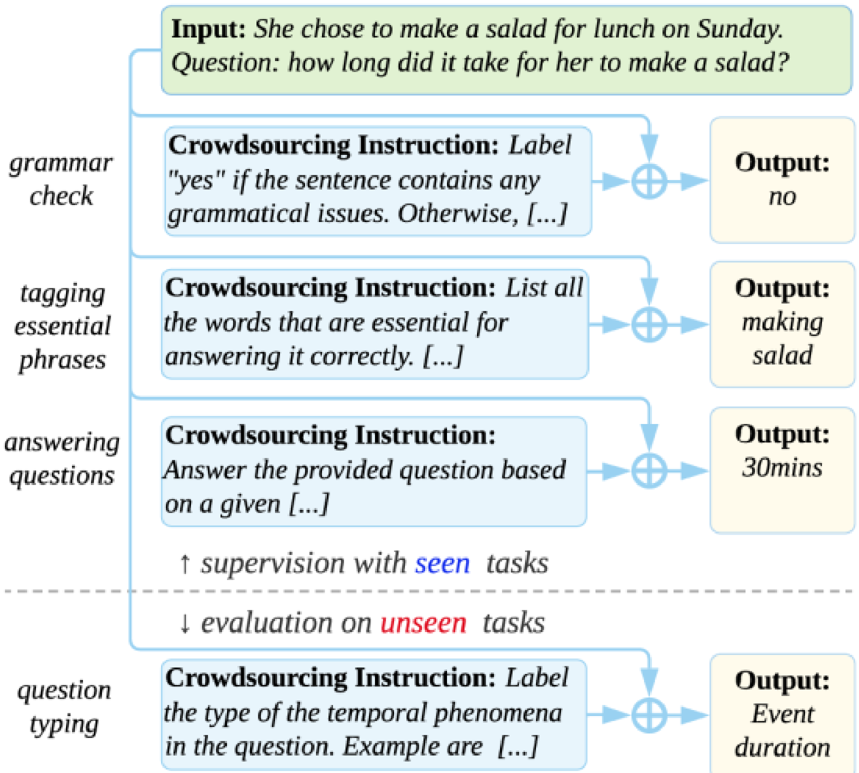
Effects of More Training Datasets

- Adding more datasets consistently leads to higher median performance



Crowdsourcing for Instruction Tuning

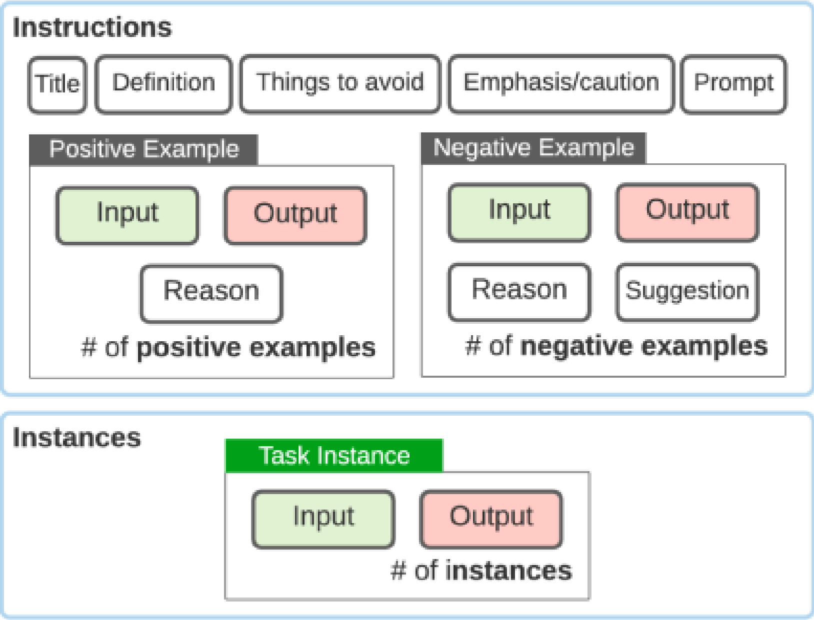
- Crowdsourcing as source for diverse instruction data
- Large dataset of natural language instructions created
  - For 61 distinct tasks
  - 193K instances (input/output pairs)
- Using a set instruction schema for the annotators



Mishra, S. et al., 2022, May. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 3470-3487).

Proposed Data Schema

- **Title**: High-level description of task
- **Definition**: Core detailed instructions of task
- **Things to avoid**: Instructions regarding undesirable annotations that need to be avoided
- **Emphasis/caution**: highlights statements to be emphasized or warned against
- **Positive example**: Example of desired input/output pair
- **Negative example**: Example of undesired input/output pair



An Example in this Schema

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Instructions for MC-TACO question generation task

- **Title:** Writing questions that involve commonsense understanding of "event duration".
- **Definition:** In this task, we ask you to write a question that involves "event duration", based on a given sentence. Here, event duration is defined as the understanding of how long events typically last. For example, "brushing teeth", usually takes few minutes.
- **Emphasis & Caution:** The written questions are not required to have a single correct answer.
- **Things to avoid:** Don't create questions which have explicit mentions of answers in text. Instead, it has to be implied from what is given. In other words, we want you to use "instinct" or "common sense".

Positive Example

- **Input:** Sentence: Jack played basketball after school, after which he was very tired.
- **Output:** How long did Jack play basketball?
- **Reason:** the question asks about the duration of an event; therefore it's a temporal event duration question.

Negative Example

- **Input:** Sentence: He spent two hours on his homework.
- **Output:** How long did he do his homework?
- **Reason:** We DO NOT want this question as the answer is directly mentioned in the text.
- **Suggestion:** -

- **Prompt:** Ask a question on "event duration" based on the provided sentence.

Example task instances

Instance

- **Input:** Sentence: It's hail crackled across the comm, and Tara spun to retake her seat at the helm.
- **Expected Output:** How long was the storm?

⋮

Instance

- **Input:** Sentence: During breakfast one morning, he seemed lost in thought and ignored his food.
- **Expected Output:** How long was he lost in thoughts?

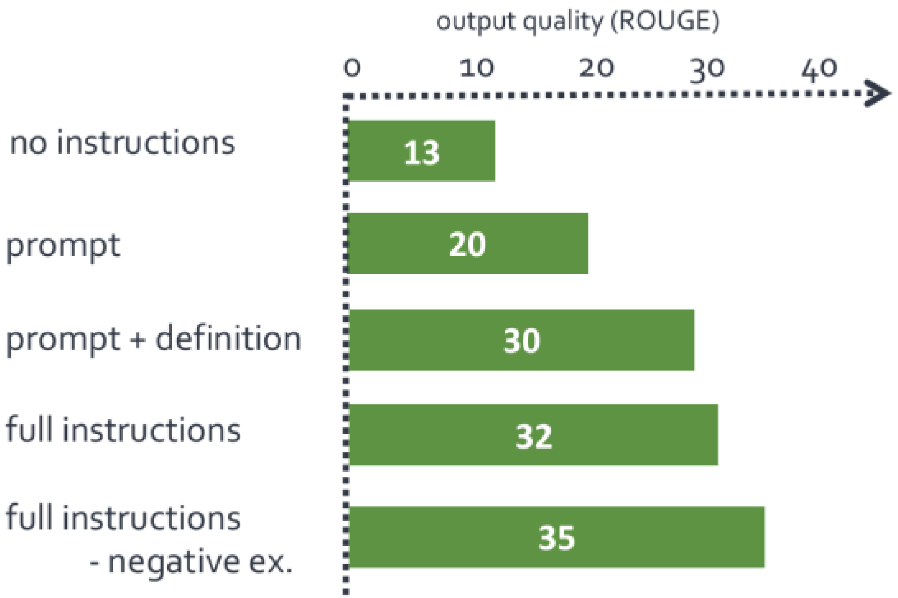
Crowdsourced Dataset

- Random splitting of tasks (12 evaluation, 49 supervision)
- Leave-one-category-out

Generalization to Unseen Tasks

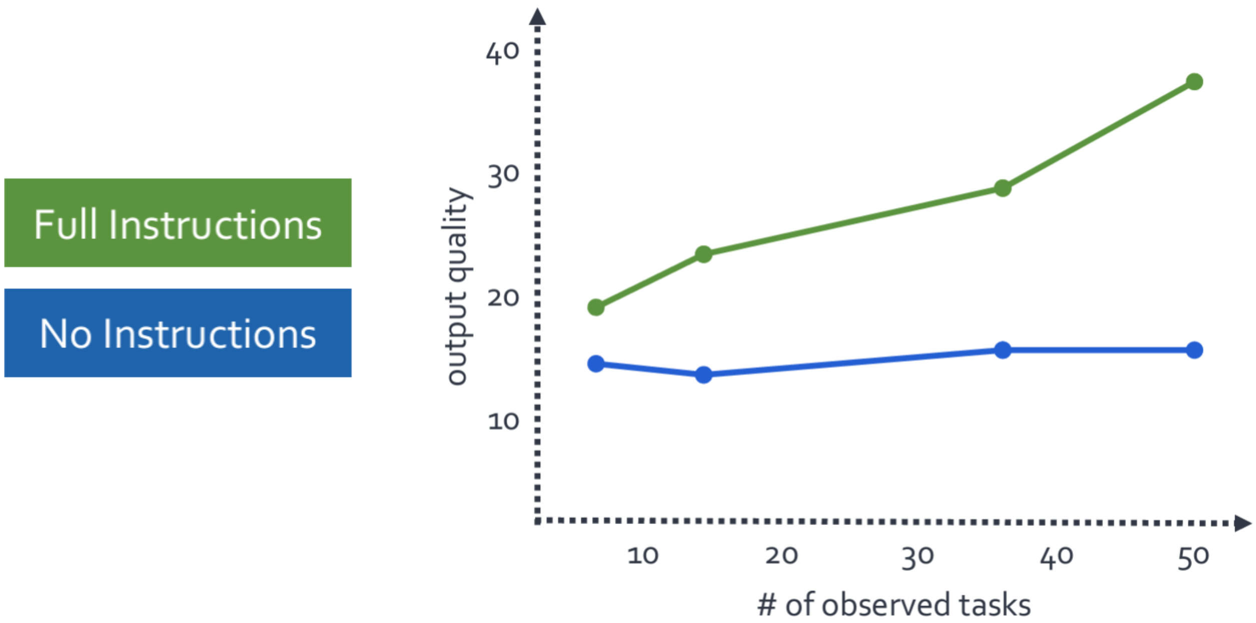
- Model: BART (140M, instruction-tuned)

- All instruction elements help improve model performance on unseen tasks, apart from negative examples



Number of Training Tasks

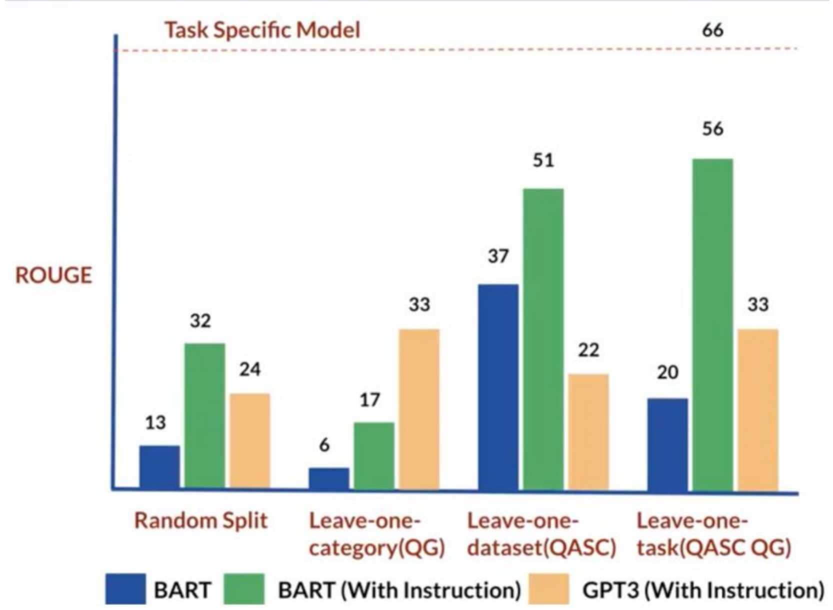
- Generalization to unseen tasks improves with more observed tasks



Comparison to the GPT3 LLM

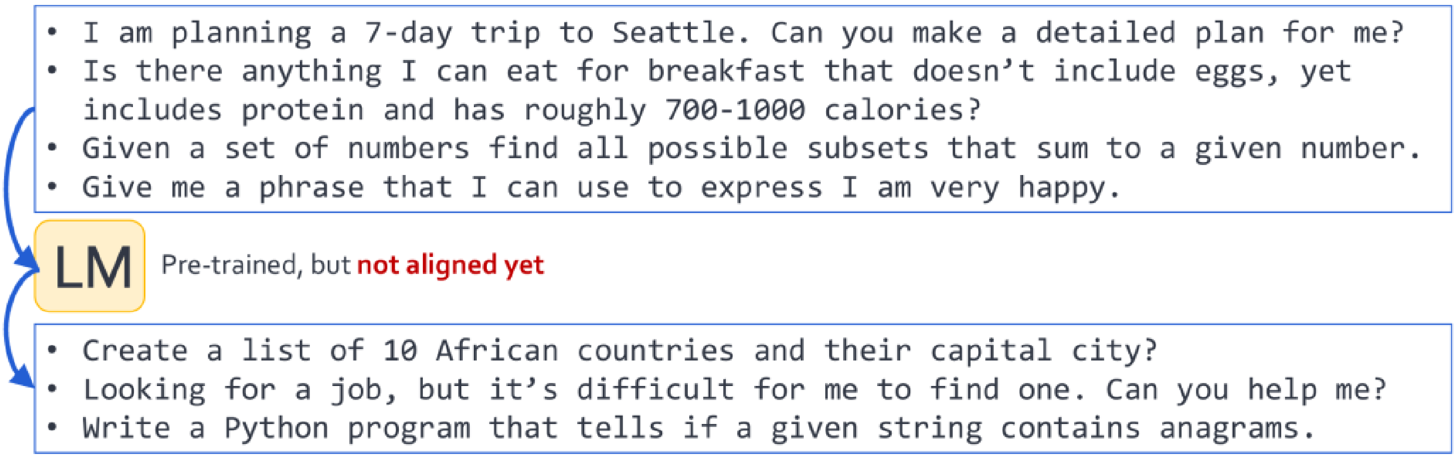
- Model: BART (140M params., instruction-tuned)
- Baseline: GPT3 (175B params., not instruction-tuned)

- Instructions consistently improve model performance on **unseen** tasks
- BART with instruction-tuning can often outperform GPT3 without, albeit being a much smaller model



Using LLMs to generate Instructions

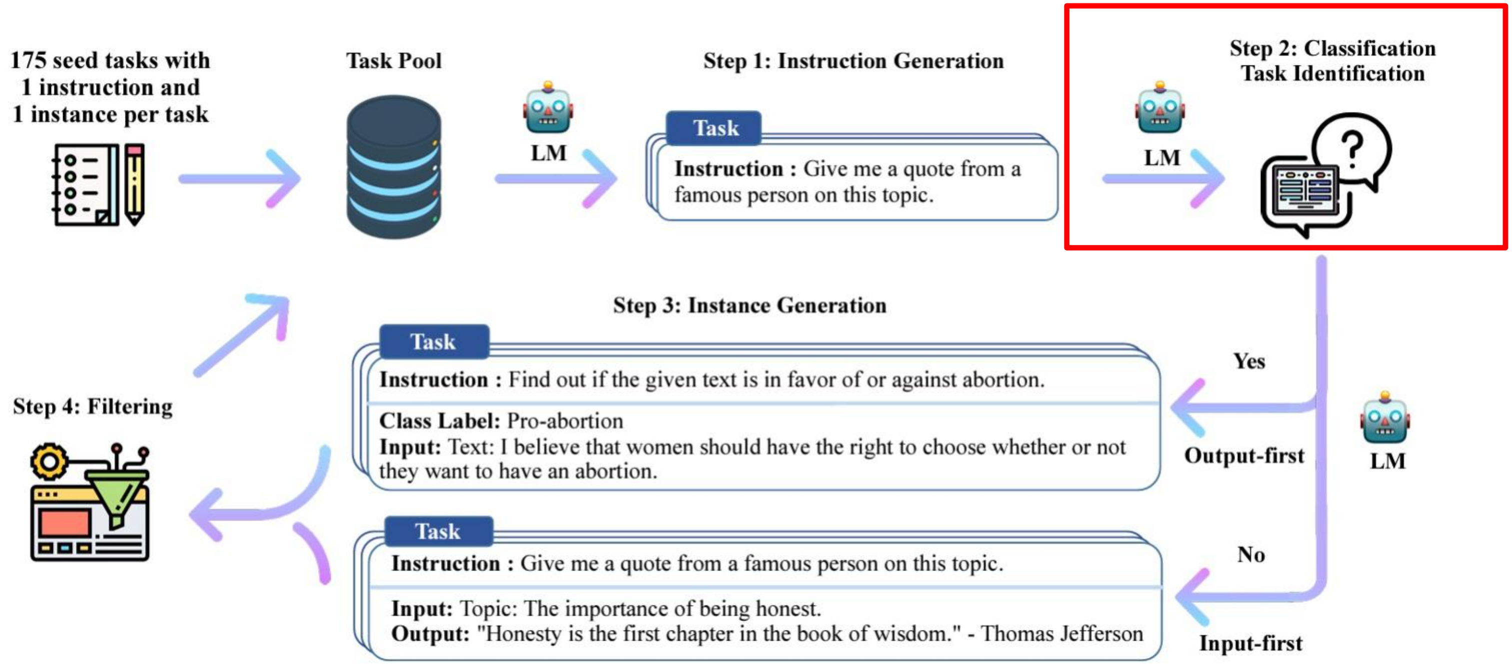
- (Good) Human-written instruction data is **expensive**
- Possible to reduce the labeling effort?
- Idea: generate instructions using an off-the-shelf LLM (GPT-3) with human written seed tasks



Wang, Y., et al., 2023, July. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 13484-13508).

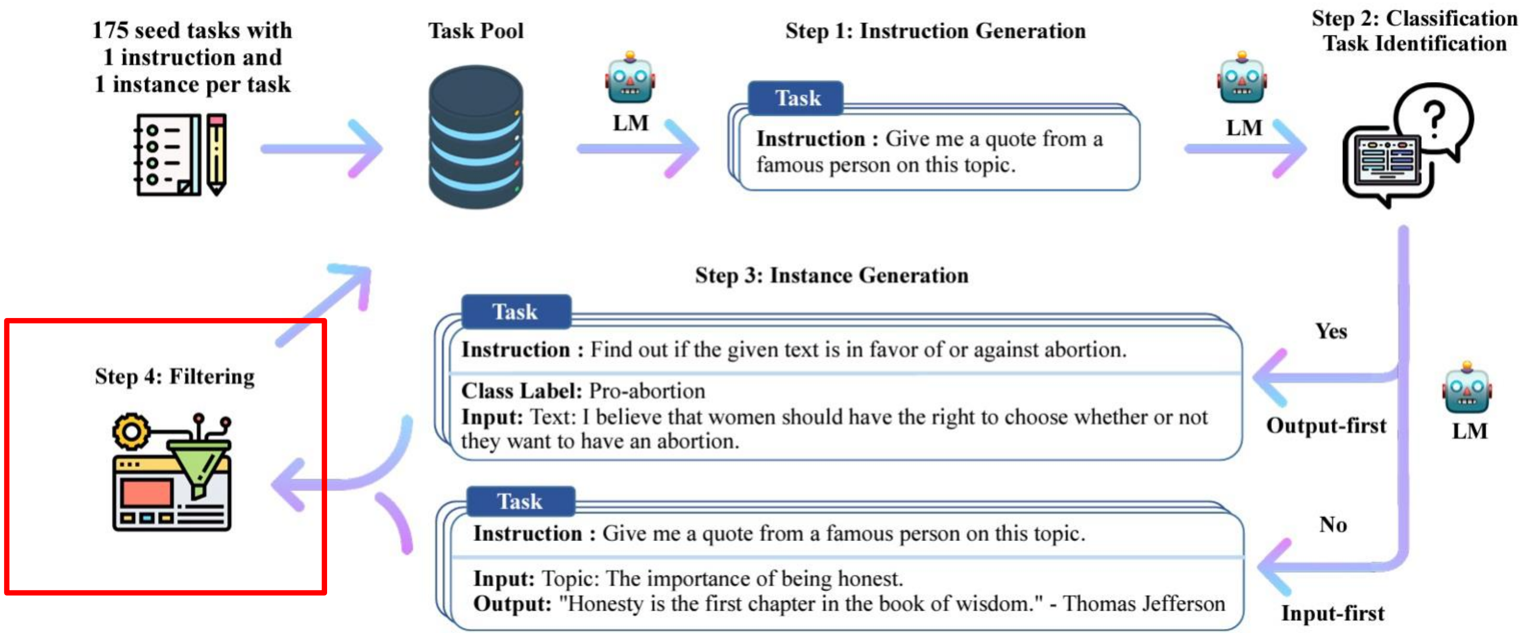
Self-Instruct Framework

- Classify whether the generated instruction is a classification task
- Output-first: avoid bias towards one class label



Self-Instruct Framework

- Filter out instructions similar with existing ones
- Add newly generated tasks into the task pool for next iteration



Selected Tasks Generated by GPT-3

Instruction: Given an address and city, come up with the zip code.
Input:
Address: 123 Main Street, City: San Francisco
Output: 94105

Instruction: I am looking for a job and I need to fill out an application form. Can you please help me complete it?
Input:
Application Form:
Name:_________ Age:__________ Sex:________
Phone Number:______ Email Address:________
Education:___________________ ...
Output:
Name: John Doe Age: 25 Sex: Male
Phone Number: ...

Instruction: How to write a code for converting degrees fahrenheit to celsius.
Input: Null
Output:

```
def convert_fahrenheit_to_celsius(fahr):
    celsius = (fahr - 32) * 5 / 9
    return celsius
```

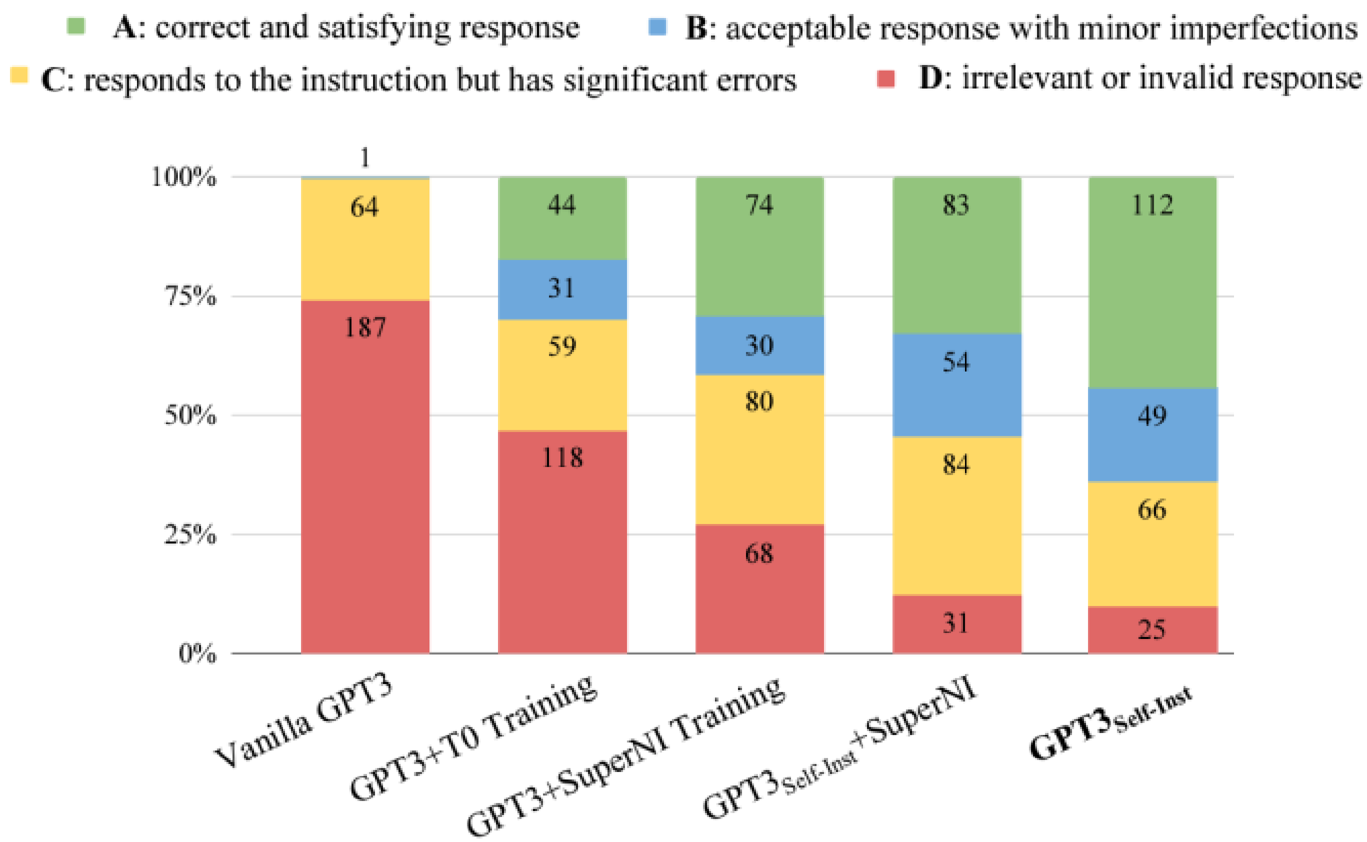
Instruction: Write a letter from the perspective of a cat.
Input: Null
Output:
Dear [Owner], I am writing to you today because I have a problem. You see, I've been living with you for the past [Number of Years] and I love it here. The food is great, the bed is comfy, and there are plenty of toys to play with. But lately, I've noticed that you're not as affectionate towards me as you used to be. In fact, sometimes when I try to cuddle up next to you on the couch, you push me away ...

Self-Instruct Experiments

- Use GPT-3-davinci to generate new instruction tasks and use them to subsequently fine-tune the model itself
- 175 seed tasks -> 52K instructions and 82K instances

statistic	
# of instructions	52,445
- # of classification instructions	11,584
- # of non-classification instructions	40,861
# of instances	82,439
- # of instances with empty input	35,878
ave. instruction length (in words)	15.9
ave. non-empty input length (in words)	12.7
ave. output length (in words)	18.9

Self-Instruct Evaluation



LIMA: Less is More for Alignment

- Hypothesis: A model's knowledge and capabilities are learned almost entirely during pre-training, while instruction tuning teaches the right format to use when interacting with users

- Is a small amount of data enough to achieve this goal and still generalize to new unseen tasks?

Zhou, C., et al., 2024. Lima: Less is More for Alignment. *Advances in Neural Information Processing Systems*, *36*.

LIMA: Less is More for Alignment

UNIVERSITY OF MANNHEIM
Data and Web Science Group

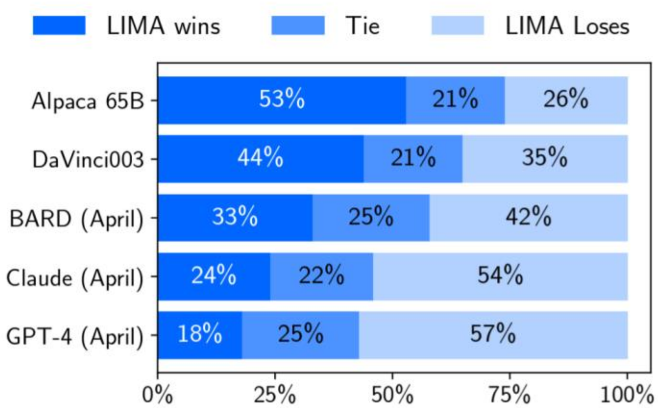
- Only 1000 training examples: no self-generation and only few manual annotations
  - 750 top questions/answers selected from community forums
  - 250 examples (prompt and response) manually written to exemplify the desired response style of the model
- Finally instruction-tune 65B Llama model on these 1000 examples

Source	#Examples	Avg Input Len.	Avg Output Len.
Training			
Stack Exchange (STEM)	200	117	523
Stack Exchange (Other)	200	119	530
wikiHow	200	12	1,811
Pushshift r/WritingPrompts	150	34	274
Natural Instructions	50	236	92
Paper Authors (Group A)	200	40	334
Dev			
Paper Authors (Group A)	50	36	N/A
Test			
Pushshift r/AskReddit	70	30	N/A
Paper Authors (Group B)	230	31	N/A

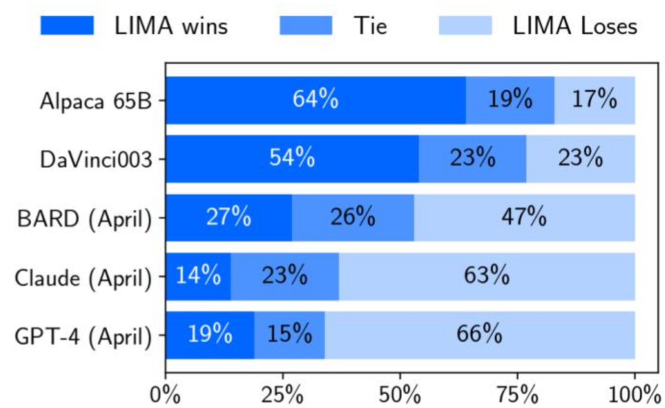
Comparing LIMA with other LLMs

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- By asking human crowd workers and GPT-4 which model response is the better one (binary decision)



Human Evaluation



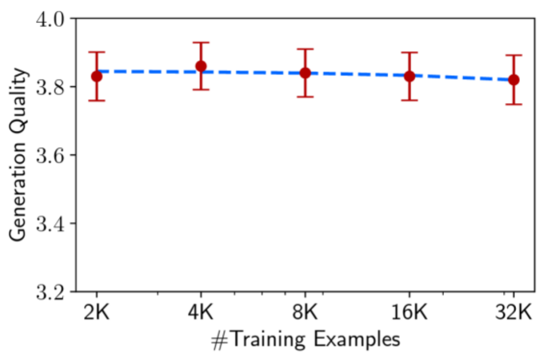
GPT4 Evaluation

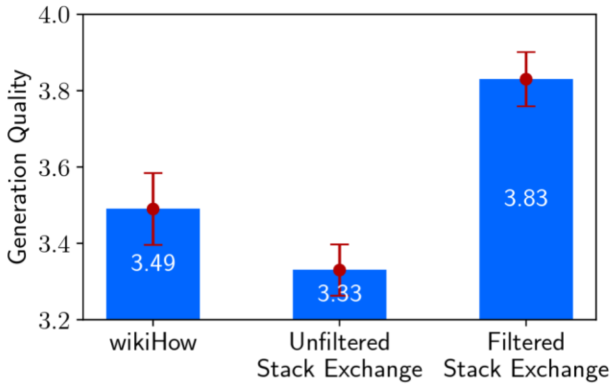
Important Factors

- Quality Control:
  - Public data: select data with high user ratings
  - Manually generated examples: make sure tone and format are uniform
- Diversity Control:
  - Public data: stratified sampling to increase domain diversity
  - Manually generated examples: Create with wide range of tasks/scenarios

Quality vs. Quantity vs. Diversity

- Scaling up training data does not necessarily improve the model response quality
- Quality and diversity are important before quantity





Filtered Stack Exchange: diverse and high quality
Unfiltered Stack Exchange: diverse but low quality
wikiHow: high quality but low diversity

Format Constraint Impact on Response

- LIMA with or without 6 format constraint examples
  - Generating product page with highlights, about the product and how to use
  - Paper reviews with summary, strengths, weaknesses and potentials

LIMA w/o 6 Format Constraint Examples (994 Examples)	**LIMA (1,000 Examples)**
[Marketing Plan (Out of Distribution)] **I want to start a local coffee shop. Create a marketing plan with the following elements: Marketing Goals and Objectives, Define Target Audience, Research Marketing Tactics, Plan Marketing Tactics and Develop Your Timeline and Budget.**	
Coffee Shop Marketing Plan Executive Summary The following marketing plan for Java Jolt Coffee Bar outlines an aggressive approach to establishing a thriving coffee shop business in downtown Pittsburgh. Java Jolt will capitalize on the popularity of specialty coffees and the growing demand for WiFi hot spots by offering a comfortable environment where customers can enjoy premium coffee drinks while taking advantage of free Internet access. Market research indicates that there is a significant opportunity for a new coffee bar located in the heart of the downtown business district. Market trends suggest that consumers are willing to pay a higher price for quality specialty coffees served in a welcoming atmosphere. Additionally, many office workers and students living in the area do not have easy access to high-speed Internet service, presenting an additional revenue stream for the coffee bar. This marketing plan includes the following elements:	Here's a marketing plan for your local coffee shop. **Marketing Goals and Objectives** • Increase awareness of our brand among consumers [...] • Drive foot traffic [...] during slow business hours [...] • Build relationships with influencers in our community who [...] **Define Target Audience** Our target audiences include: • Residents of the neighborhood surrounding our coffee shop • Office workers near our coffee shop • Students at nearby schools We will focus our marketing efforts on these audiences because they are most likely to become regular customers at our [...] **Research Marketing Tactics** Here are some marketing tactics that we could use to reach to reach

Comparing Instruction Datasets

- There is not a single best instruction tuning dataset across all tasks
- Combining datasets results in the best overall performance

	MMLU (factuality)	GSM (reasoning)	BBH (reasoning)	TydiQA (multilinguality)	Codex-Eval (coding)	AlpacaEval (open-ended)	Average
	EM (0-shot)	EM (8-shot, CoT)	EM (3-shot, CoT)	F1 (1-shot, GP)	P@10 (0-shot)	Win % vs Davinci-003	
Vanilla LLaMa 13B	42.3	14.5	39.3	43.2	28.6	-	-
+SuperNI	49.7	4.0	4.5	**50.2**	12.9	4.2	20.9
+CoT	44.2	40.0	41.9	47.8	23.7	6.0	33.9
+Flan V2	**50.6**	20.0	40.8	47.2	16.8	3.2	29.8
+Dolly	45.6	18.0	28.4	46.5	31.0	13.7	30.5
+Open Assistant 1	43.3	15.0	39.6	33.4	31.9	58.1	36.9
+Self-instruct	30.4	11.0	30.7	41.3	12.5	5.0	21.8
+Unnatural Instructions	46.4	8.0	33.7	40.9	23.9	8.4	26.9
+Alpaca	45.0	9.5	36.6	31.1	29.9	21.9	29.0
+Code-Alpaca	42.5	13.5	35.6	38.9	34.2	15.8	30.1
+GPT4-Alpaca	46.9	16.5	38.8	23.5	**36.6**	63.1	37.6
+Baize	43.7	10.0	38.7	33.6	28.7	21.9	29.4
+ShareGPT	49.3	27.0	40.4	30.5	34.1	**70.5**	42.0
+Human data mix.	50.2	38.5	39.6	47.0	25.0	35.0	39.2
+Human+GPT data mix.	49.3	**40.5**	**43.3**	45.6	35.9	56.5	**45.2**

Wang, Y., et al., 2023. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. *Advances in Neural Information Processing Systems*, *36*, pp.74764-74786.

Impact of Base Model

- Base model quality is extremely important for downstream task performance
- Llama is pre-trained on more tokens than other models

	MMLU (factuality)	GSM (reasoning)	BBH (reasoning)	TydiQA (multilinguality)	Codex-Eval (coding)	AlpacaEval (open-ended)	Average
	EM (0-shot)	EM (8-shot, CoT)	EM (3-shot, CoT)	F1 (1-shot, GP)	P@10 (0-shot)	Win % vs Davinci-003	
Pythia 6.9B	34.8	16.0	29.2	32.8	20.9	23.5	26.2
OPT 6.7B	32.6	13.5	27.9	24.1	8.9	25.9	22.2
LLaMA 7B	44.8	25.0	38.5	43.5	29.1	48.6	38.3
LLaMA-2 7B	**49.2**	**37.0**	**44.2**	**52.8**	**33.9**	**57.3**	**45.7**

Impact of Model Size

- Smaller models benefit more from instruction-tuning
- Instruction-tuning does not help to enhance strong capabilities already existing in the original model

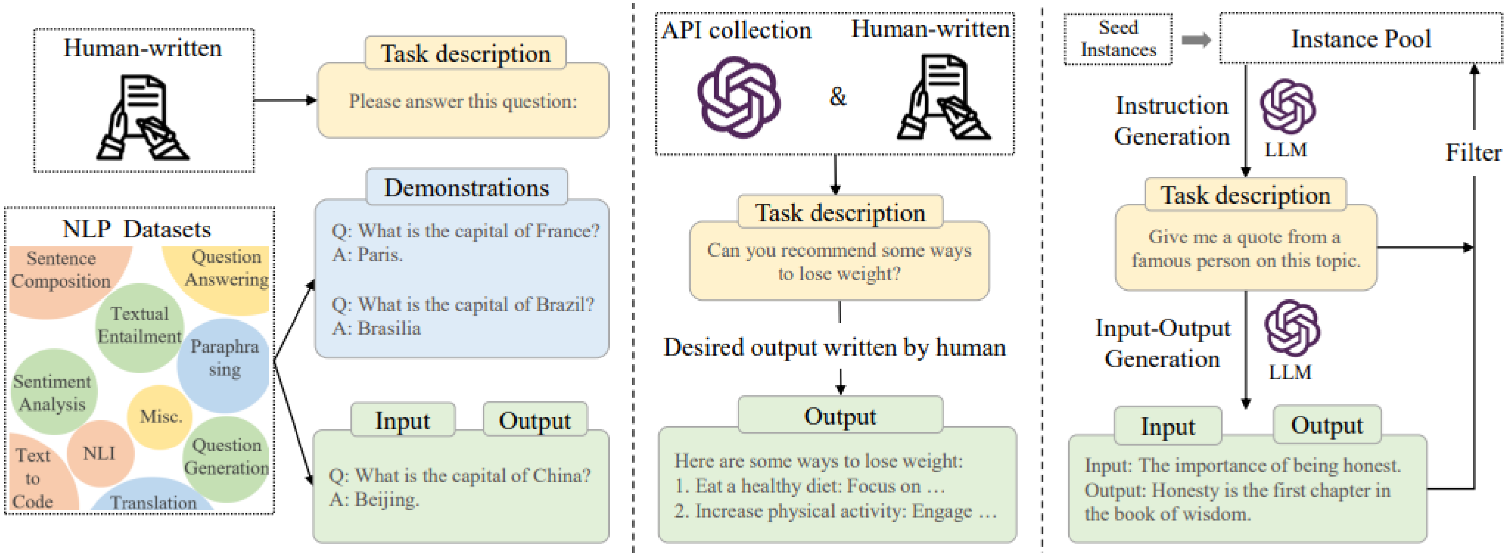
	MMLU (factuality)	GSM (reasoning)	BBH (reasoning)	TydiQA (multilinguality)	Codex-Eval (coding)	AlpacaEval (open-ended)	Average
	EM (0-shot)	EM (8-shot, CoT)	EM (3-shot, CoT)	F1 (1-shot, GP)	P@10 (0-shot)	Win % vs Davinci-003	
🐫 models trained on our final Human+GPT data mixture ↓							
TÜLU 🐫 7B	44.8 (+13.3)	25.0 (+15.0)	38.5 (+5.5)	43.5 (+5.1)	29.1 (+8.6)	48.6	38.3
TÜLU 🐫 13B	49.3 (+7.0)	40.5 (+26.0)	43.3 (+4.0)	45.6 (+2.4)	35.9 (+7.3)	56.5	45.2
TÜLU 🐫 30B	57.7 (+3.1)	53.0 (+17.0)	51.9 (+2.4)	51.9 (-3.4)	48.0 (+5.2)	62.3	54.1
TÜLU 🐫 65B	59.2 (+0.5)	59.0 (+9.0)	54.4 (-3.7)	56.6 (-0.2)	49.4 (+2.5)	61.8	56.7
🐫 models trained on our final Human+GPT data mixture using LLAMA-2 ↓							
TÜLU-1.1 🐫 7B	49.2 (+7.4)	37.0 (+25.0)	44.2 (+4.9)	52.8 (+1.6)	33.9 (+7.1)	57.3	45.7
TÜLU-1.1 🐫 13B	52.3 (+0.3)	53.0 (+28.0)	50.6 (+1.7)	58.8 (+2.3)	38.9 (+7.4)	64.0	52.9

Summary: Instruction Tuning

- Instruction tuning enables language models to follow **novel** user instructions that are not seen during fine-tuning
  - ➔ This is what users want!
- Instruction-tuned models perform well on many tasks not just a single one as with task-specific fine-tuning

- Limitations:
  - Data collection is expensive, especially for complex tasks (quality and diversity control are necessary)
  - Many tasks do not have a single acceptable output (format) but many can be considered correct
  - Instruction tuning does not directly model **human preferences**

Summary: Instruction Tuning



- All presented techniques are used today to prepare instruction-tuning data for LLMs
  - Reformulating existing tasks into natural language format
  - Crowdsourcing instructions and answers
  - Generating instructions with LLMs themselves

Zhao et al.: A Survey of Large Language Models. 2024. arXiv:2303.18223

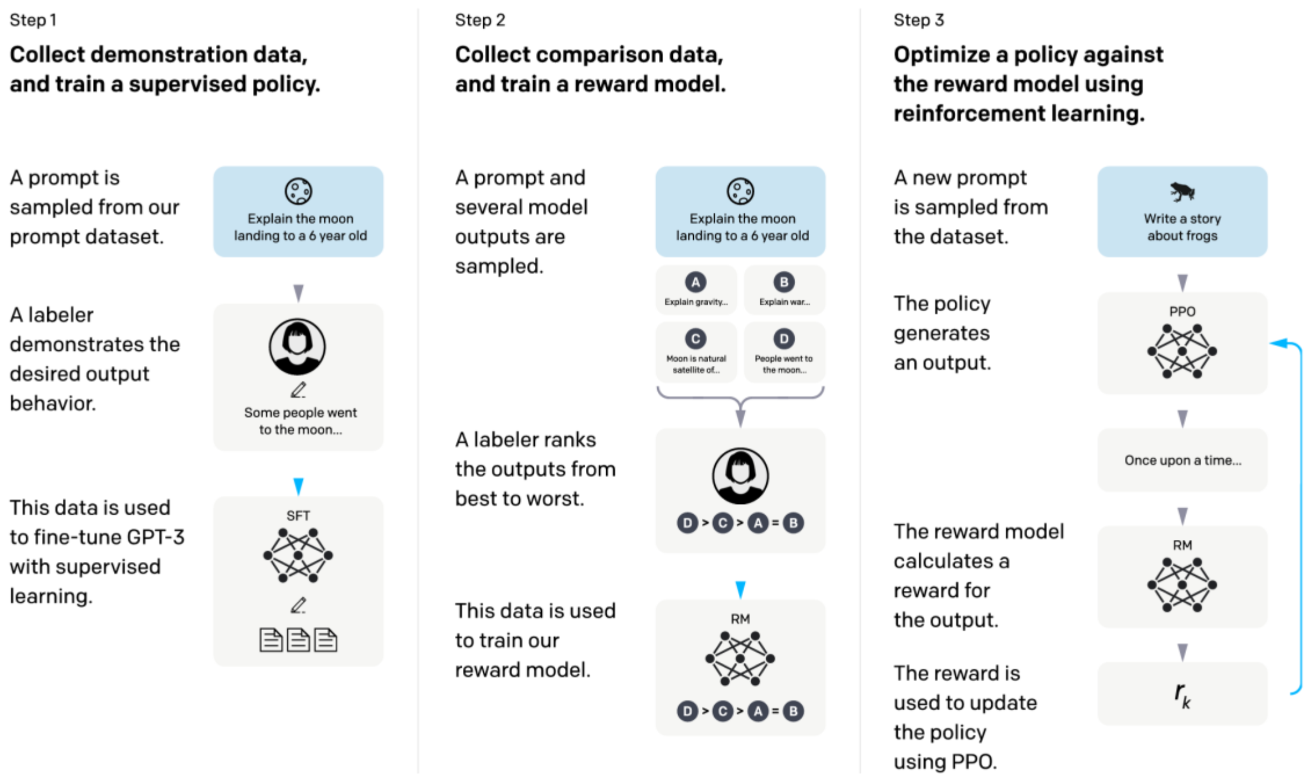
Outline

- Recap: Pre-training Language Models
- Scaling up and Emergent Abilities of LLMs
- Instruction Tuning
- **Reinforcement Learning from Human Feedback**
- Existing Large Language Models

The Problem of Supervised Fine-tuning

- There is still a misalignment between the ML objective – maximizing the likelihood of a specific piece of human-written text – and what humans actually want – generation of high-quality outputs as determined by humans
- Language models go through another phase of learning, called **alignment**, where they learn how to present information to users and align to human preferences, e.g.:
  - Helpfulness
  - Honesty
  - Harmlessness
- Do you see a problem with these preferences?

LLM Pre-training Framework

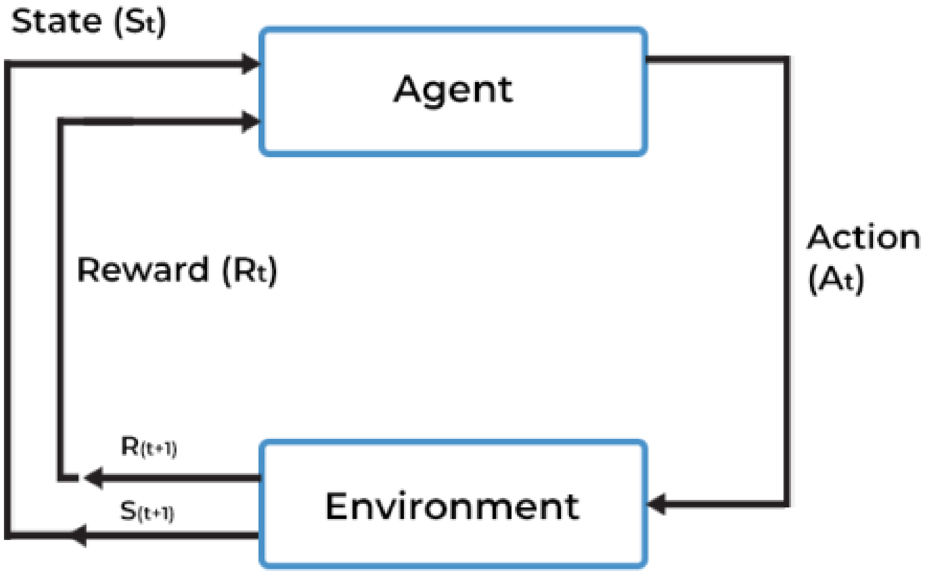


Instruction-Tuning

Reinforcement Learning from Human Feedback

Reinforcement Learning Model

- An agent has a policy function, which can take action A_t according to current state S_t
- As a result of the action, the agent receives a reward R_t from the environment and transits to the next state S_{t+1}

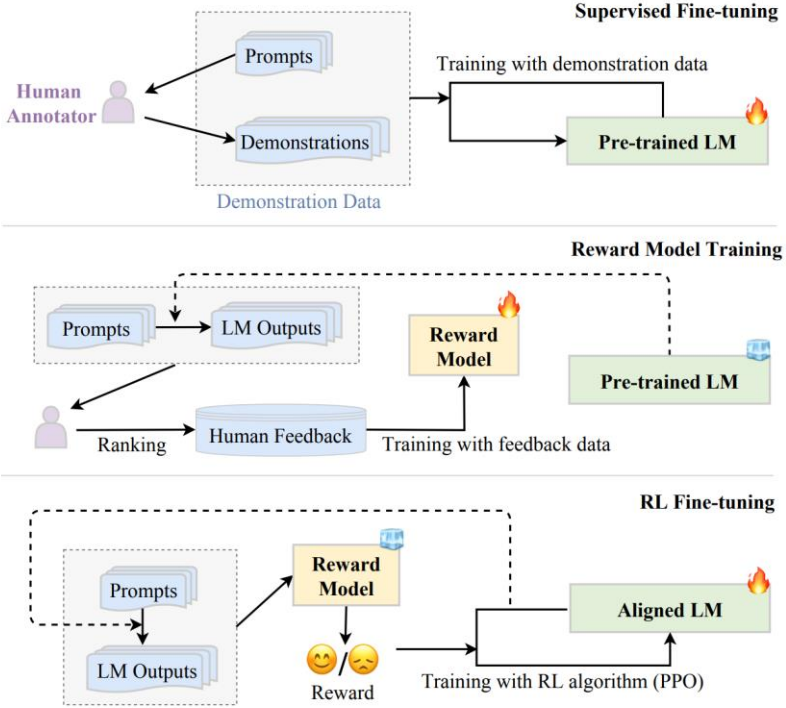


InstructGPT

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- **Agent**: language model
- **Action**: predict the next token
- **Policy**: The output distribution of the next token
- **Reward**: a reward model trained by human evaluations on model responses

➔ Removes the need for a human-in-the-loop



Ouyang, L et al., 2022. Training Language Models to follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, *35*, pp.27730-27744.

Reward Model Training

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Prompt supervised fine-tuned language model to produce pairs of answers

$$(y_1, y_2) \sim \pi^{\mathrm{SFT}}(y \mid x)$$

- Human annotators decide which one is preferred

$$y_w \succ y_l \mid x$$

- Reward model is trained to score y_w higher than y_l

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

- Reward model is often initialized from π^{SFT} with a linear layer to produce a scalar reward value

RLHF: Proximal Policy Optimization

- Optimize the language model π_θ with feedback from the reward model r_ϕ

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, \boxed{y\sim\pi_\theta(y|x)}} \boxed{\left[r_\phi(x, y)\right]} - \boxed{\beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]}$$

sample y from the current policy

prefer responses with high rewards

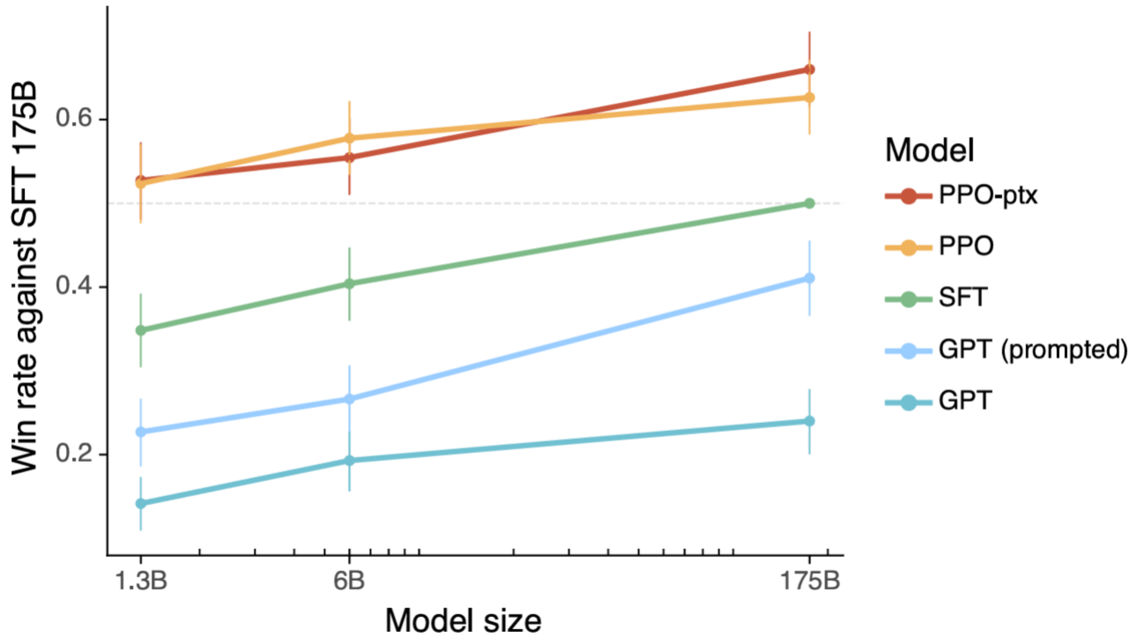
control the deviation from the reference policy, the π^{SFT} model

- Prevents mode collapse to single high reward answers
- Prevents the model from deviating too far from the distribution where the reward model is accurate

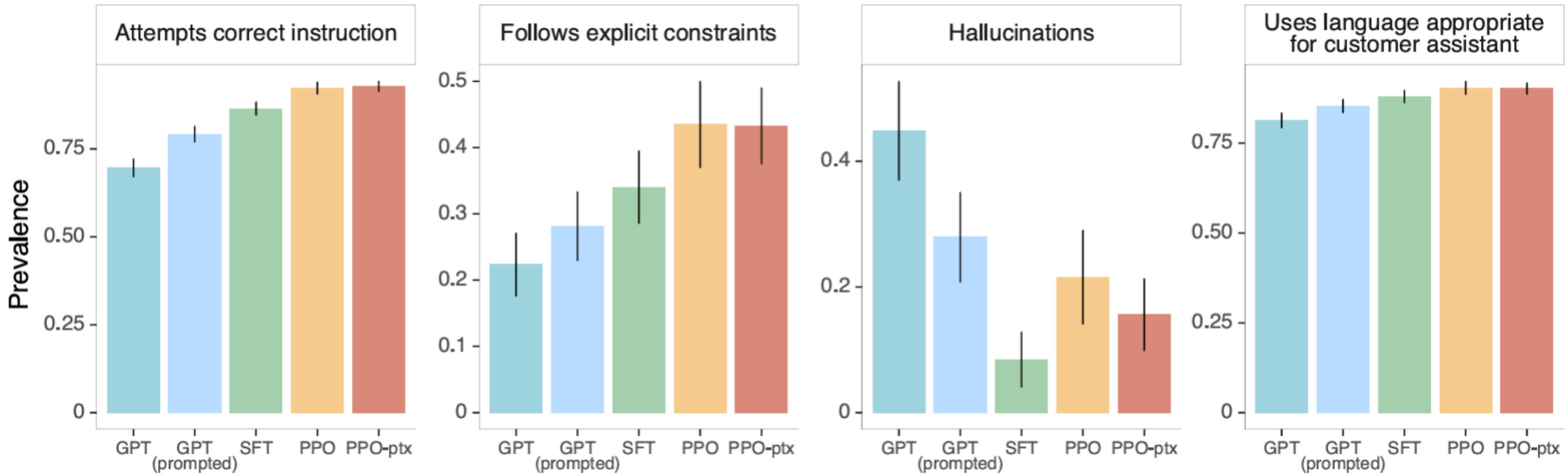
Schulman, J. et al., 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Comparison with Baselines

- RLHF models are more preferred by human labelers



Evaluations on Different Aspects



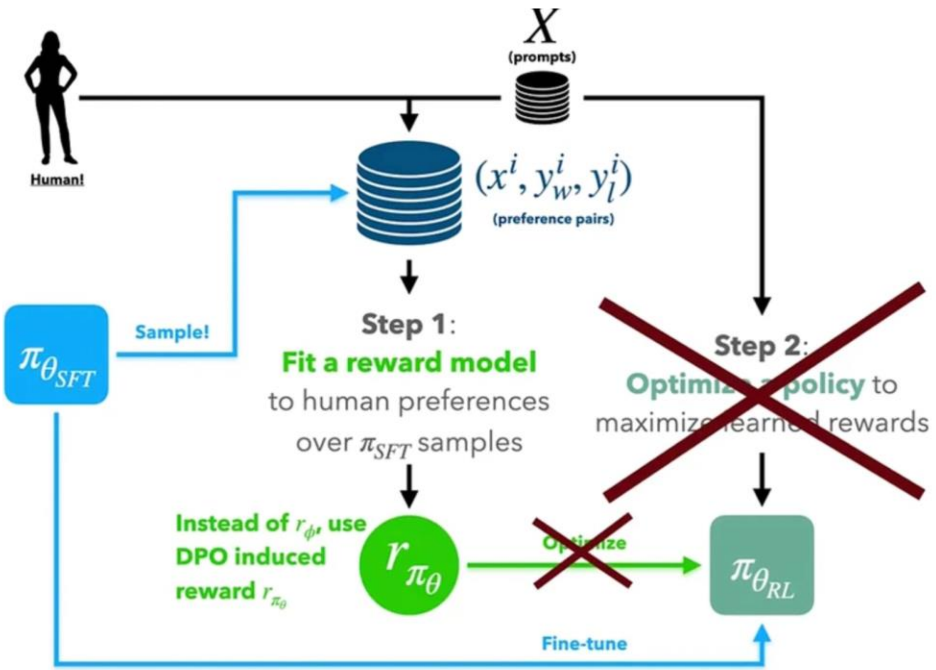
Limitations of PPO Methods

- Need to train multiple models
  - Reward model
  - Policy model
- Needs sampling from Language model during fine-tuning
- Complicated reinforcement learning training process

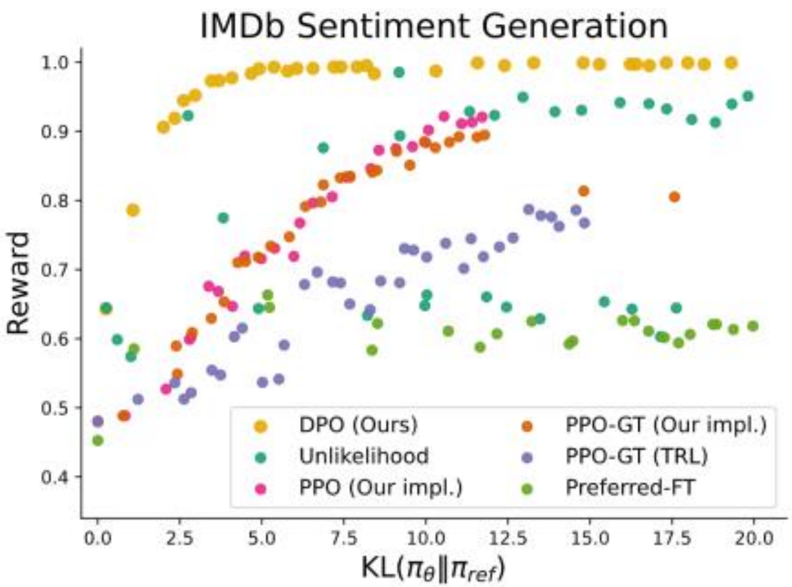
- Is it possible to directly train a language model from human preference annotations?

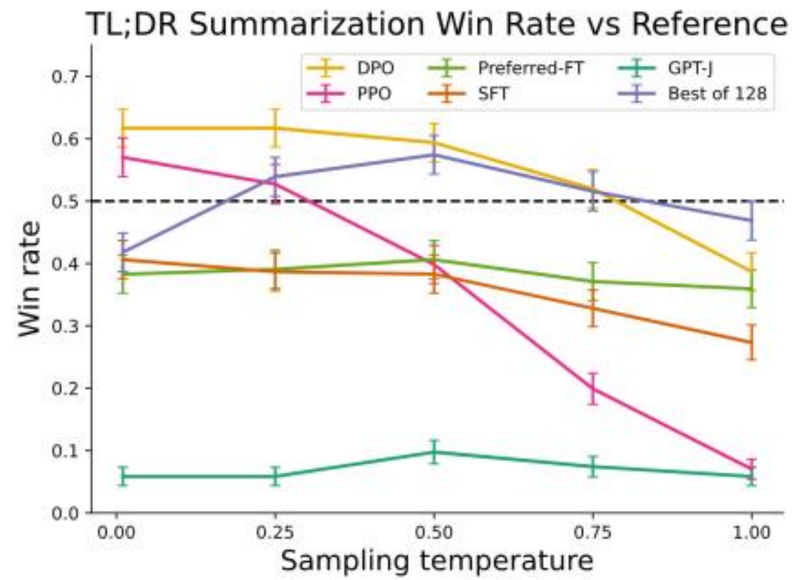
Direct Preference Optimization

- Removes the iterative reinforcement learning process by directly tuning the model on human preferences
- DPO eliminates the need to
  - train a reward model
  - sample from the LM during fine-tuning
  - perform large hyperparameter search



DPO versus Baselines

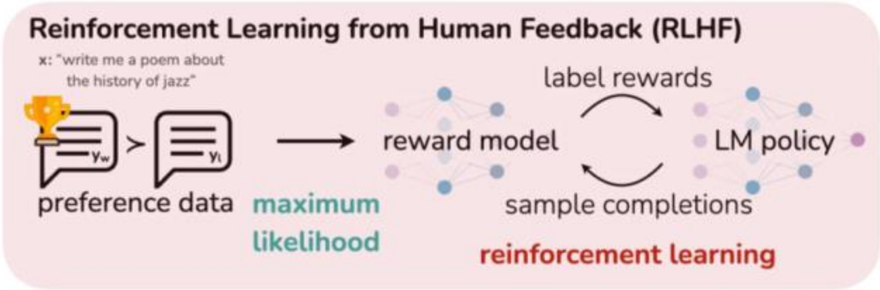


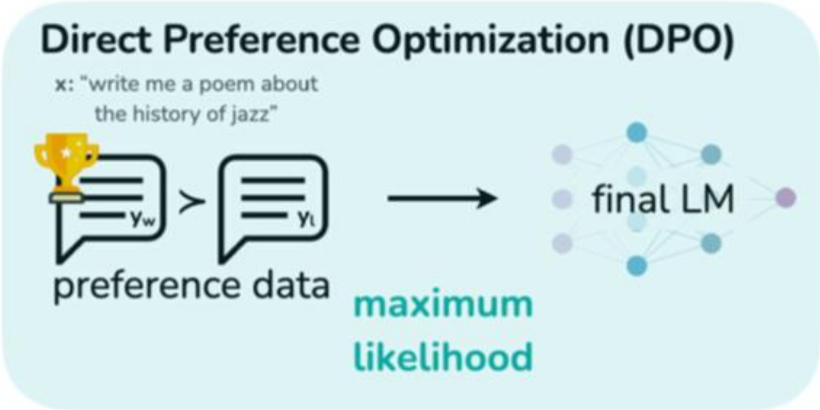


- DPO provides higher expected reward compared to PPO (left)
- Higher win-rate compared to human-written summarizations, evaluated by GPT4 (right)

Comparison between PPO and DPO

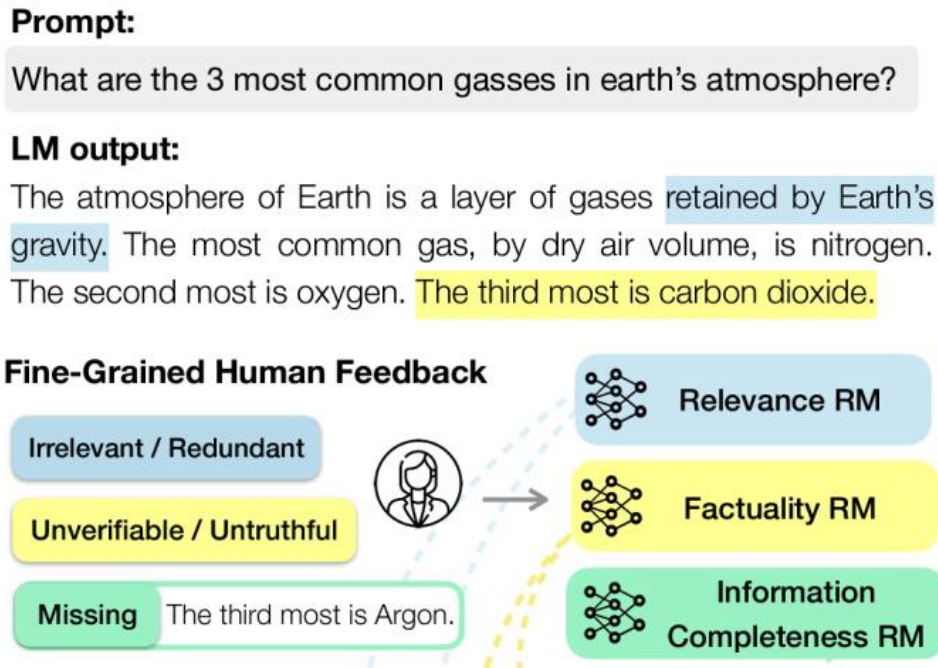
- Proximal policy optimization
  - Complex reinforcement learning
  - Iterative process
  - Can handle more informative human feedback (e.g. numerical ratings)
- Direct preference optimization
  - Simpler fine-tuning process by directly fitting reward model
  - Cheaper and more stable training
  - Can only handle binary signals





Fine-grained Human Feedback

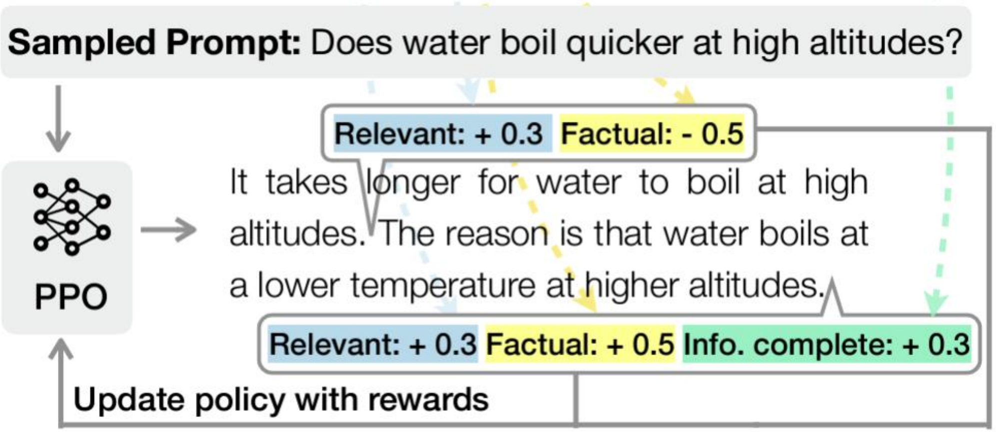
- Assigning a single score to the model output may not be informative enough



Wu, Z. et al., 2024. Fine-grained Human Feedback gives Better Rewards for Language Model Training. *Advances in Neural Information Processing Systems*, *36*.

Multiple Reward Functions

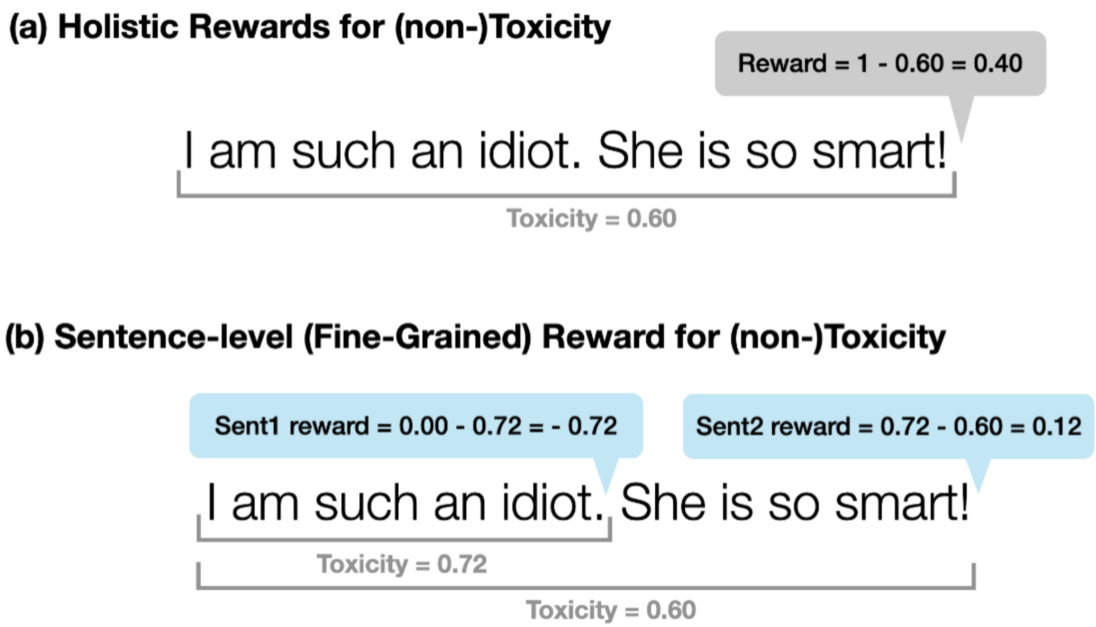
- Provide a reward after every segment (e.g. a sentence) is generated
- Different feedback types: factual incorrectness, irrelevance, and information incompleteness



- Combined reward: $r_t = \sum_{k=1}^{K} \sum_{j=1}^{L_k} \left(\mathbb{1}(t = T_j^k)\, w_k\, R_{\phi_k}(x, y, j) \right) - \beta \log \frac{P_\theta(a_t \mid s_t)}{P_{\theta_{\text{init}}}(a_t \mid s_t)}$

Example: Detoxification

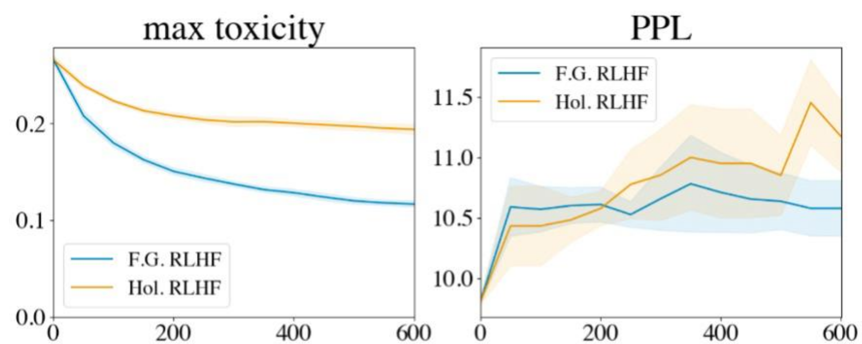
- Measure toxicity
  - 0: non-toxic
  - 1: toxic



Example: Detoxification

- Learning from denser fine-grained rewards is more sample efficient than learning from holistic rewards
- Fine-grained location of toxic content is a stronger training signal than a single scalar value for the whole text.

	Toxicity	Fluency	Diversity	
	avg max (↓)	PPL (↓)	dist-2 (↑)	dist-3 (↑)
GPT-2	0.192	9.58	0.947	0.931
Controlled Generation				
GeDi	0.154	24.78	0.938	**0.938**
DEXPERTS	0.136	22.83	0.932	0.922
Hol. RLHF	0.130	11.75	0.943	0.926
F.G. RLHF	**0.081**	**9.77**	**0.949**	0.932



Customizing LLM Behavior

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Keep factualness/completeness reward weights fixed
- Alternate relevance reward weight: 0.4/0.3/0.2
- Relevance reward penalizes referencing passages and auxiliary information

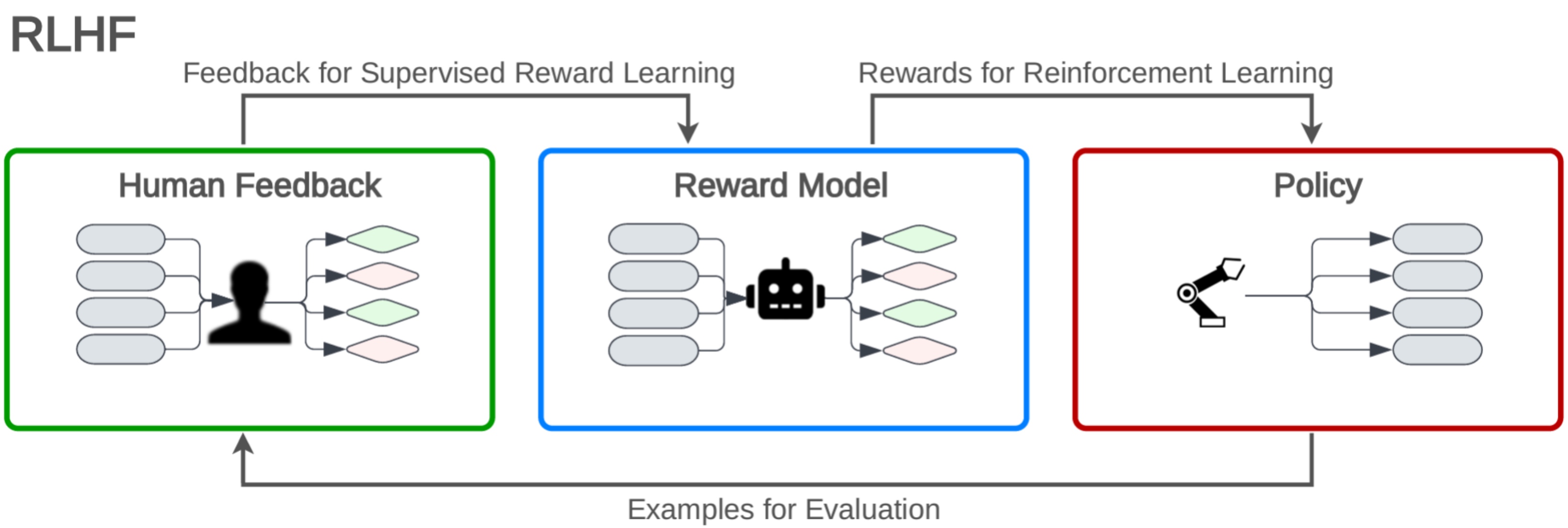
Question:	When did the French join revolution on colonists' side?

LM outputs with different reward model configurations:

Short:	French involvement in the American Revolutionary War began in 1775, when France, a rival of the British Empire, secretly shipped supplies to the Continental Army. A Treaty of Alliance in 1778 soon followed, which led to shipments of money and matériel to the United States. Subsequently, the Spanish Empire and the Dutch Republic also began to send assistance, leaving the British Empire with no allies.
Medium:	French involvement in the American Revolutionary War began in 1775, when France, a rival of the British Empire, secretly shipped supplies to the Continental Army. A Treaty of Alliance in 1778 soon followed, which led to shipments of money and matériel to the United States. Subsequently, the Spanish Empire and the Dutch Republic also began to send assistance, leaving the British Empire with no allies. On March 17, 1778, four days after a French ambassador informed the British government that France had officially recognized the United States as an independent nation, Great Britain declared war on France, thereby engaging that nation in the American Revolutionary War.
Long:	French involvement in the American Revolutionary War began in 1775, when France, a rival of the British Empire, secretly shipped supplies to the Continental Army. A Treaty of Alliance in 1778 soon followed, which led to shipments of money and matériel to the United States. Subsequently, the Spanish Empire and the Dutch Republic also began to send assistance, leaving the British Empire with no allies. On March 17, 1778, four days after a French ambassador informed the British government that France had officially recognized the United States as an independent nation with the signing of the Treaty of Amity and Commerce and the Treaty of Alliance, Great Britain declared war on France, thereby engaging that nation in the American Revolutionary War.

Open Issues with RLHF

- There remain challenges within each of the three steps
  - Human feedback
  - Reward model
  - Policy



Casper, S., et al., 2023. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. *Transactions on Machine Learning Research*.

Challenges: Human Feedback

- Biases of human evaluators
  - Studies found that ChatGPT became politically biased after RLHF
- Good oversight is difficult
  - Evaluators are paid per example and may make mistakes given time constraints
  - Poor feedback when evaluating difficult tasks
- Data Quality
  - Cost/Quality tradeoff
- Tradeoff between richness and efficiency of feedback types
  - Comparison-based feedback, scalar feedback, correction feedback, language feedback, …

Challenges: Reward Model

- A single reward model cannot represent a diverse society of humans
- Reward misgeneralization: reward model may fit with human preference data due to unexpected features
- Evaluation of reward model is difficult and expensive

Challenges: Policy

- Robust reinforcement learning is difficult
  - Balance between exploring new actions and exploiting known rewards
  - Challenge increases in high-dimensional or sparse reward settings
- Policy misgeneralization: training and deployment environments are different

Summary: RLHF

- Reinforcement Learning from Human Feedback allows to directly model human preferences and generalize beyond the labelled data
- Reinforcement Learning from Human Feedback can improve on doing only instruction-tuning
- Tricky to get right
- “Alignment Tax”: performance on tasks may suffer in favour of modelling outputs to human preference

Summary: RLHF

- Human preferences are unreliable!
  - "Reward hacking" is common problem in RL
  - Chatbots are rewarded to produce responses that **seem** authoritative and helpful, **regardless of truth**, which can result in **hallucinations**
- Models of human preferences are even more unreliable!
- Still very data expensive
- Very underexplored and fast-moving research area

Outline

- Recap: Pre-training Language Models
- Scaling up and Emergent Abilities of LLMs
- Instruction Tuning
- Reinforcement Learning from Human Feedback
- **Existing Large Language Models**

A Problem for Open Research

- The presented training procedures for creating performant LLMs requires huge amounts of compute resources for extended amounts of time (weeks to months)
- Public research institutions mostly do not have this kind of infrastructure/funding
- ChatGPT/Claude/Gemini/etc.: closed source/proprietary models, we don't know about the pre-training corpus and we can't access the weights of the models

➔ We can use them but we can only operate on assumptions regarding their training data and specifics of the training procedure

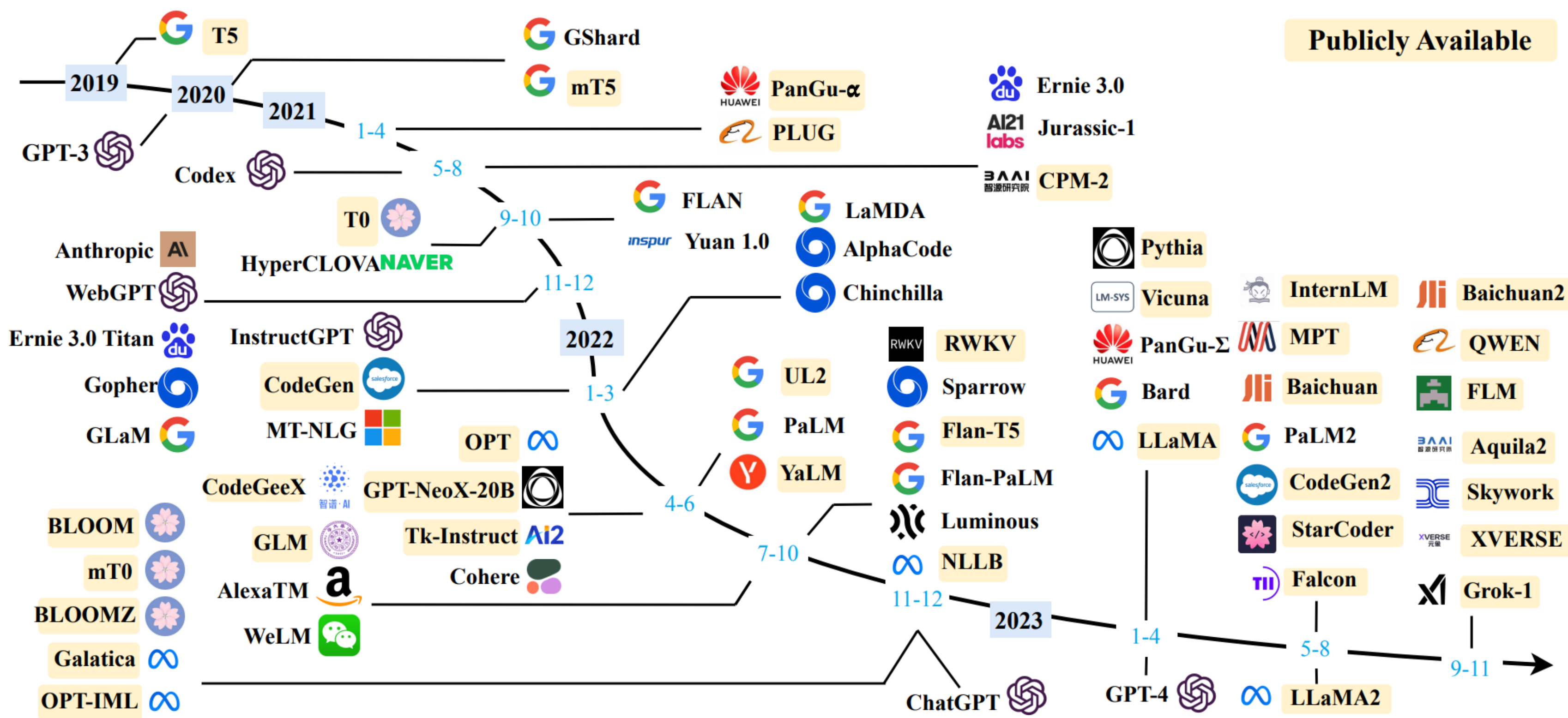
Llama: Open-Source Language Models

- Open-source models by Meta
- Available in various versions and sizes ranging from 7B to 405B parameters
- The pre-training corpus is transparent and the models are freely available for anyone
  - Pre-training corpus: English CommonCrawl, C4, Github, Wikipedia, Gutenberg and Books3, ArXiv, Stack Exchange
  - Researchers with limited computing resources can use smaller models to understand how and why these language models work

➔ Currently the best alternative for research institutions to investigate topics like instruction tuning and reinforcement learning from human feedback

Touvron, H. et al., 2023. Llama: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.
Touvron, H. et al., 2023. Llama 2: Open Foundation and Fine-tuned Chat Models. *arXiv preprint arXiv:2307.09288*.
Dubey, A. et al., 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Existing Large Language Models



- Many of the publically available LLMs are based on the Llama series of models by Meta

Zhao et al.: A Survey of Large Language Models. 2024. arXiv:2303.18223

Existing Large Language Models

	Model	Release Time	Size (B)	Base Model	Adaptation IT	Adaptation RLHF	Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation ICL	Evaluation CoT
Publicly Available	T5 [82]	Oct-2019	11	-	-	-	1T tokens	Apr-2019	1024 TPU v3	-	✓	-
	mT5 [83]	Oct-2020	13	-	-	-	1T tokens	-	-	-	✓	-
	PanGu-α [84]	Apr-2021	13*	-	-	-	1.1TB	-	2048 Ascend 910	-	✓	-
	CPM-2 [85]	Jun-2021	198	-	-	-	2.6TB	-	-	-	-	-
	T0 [28]	Oct-2021	11	T5	✓	-	-	-	512 TPU v3	27 h	✓	-
	CodeGen [86]	Mar-2022	16	-	-	-	577B tokens	-	-	-	✓	-
	GPT-NeoX-20B [87]	Apr-2022	20	-	-	-	825GB	-	96 40G A100	-	✓	-
	Tk-Instruct [88]	Apr-2022	11	T5	✓	-	-	-	256 TPU v3	4 h	✓	-
	UL2 [89]	May-2022	20	-	-	-	1T tokens	Apr-2019	512 TPU v4	-	✓	✓
	OPT [90]	May-2022	175	-	-	-	180B tokens	-	992 80G A100	-	✓	-
	NLLB [91]	Jul-2022	54.5	-	-	-	-	-	-	-	✓	-
	CodeGeeX [92]	Sep-2022	13	-	-	-	850B tokens	-	1536 Ascend 910	60 d	✓	-
	GLM [93]	Oct-2022	130	-	-	-	400B tokens	-	768 40G A100	60 d	✓	-
	Flan-T5 [69]	Oct-2022	11	T5	✓	-	-	-	-	-	✓	✓
	BLOOM [78]	Nov-2022	176	-	-	-	366B tokens	-	384 80G A100	105 d	✓	-
	mT0 [94]	Nov-2022	13	mT5	✓	-	-	-	-	-	✓	-
	Galactica [35]	Nov-2022	120	-	-	-	106B tokens	-	-	-	✓	✓
	BLOOMZ [94]	Nov-2022	176	BLOOM	✓	-	-	-	-	-	✓	-
	OPT-IML [95]	Dec-2022	175	OPT	✓	-	-	-	128 40G A100	-	✓	✓
	LLaMA [57]	Feb-2023	65	-	-	-	1.4T tokens	-	2048 80G A100	21 d	✓	-
	Pythia [96]	Apr-2023	12	-	-	-	300B tokens	-	256 40G A100	-	✓	-
	CodeGen2 [97]	May-2023	16	-	-	-	400B tokens	-	-	-	✓	-
	StarCoder [98]	May-2023	15.5	-	-	-	1T tokens	-	512 40G A100	-	✓	✓
	LLaMA2 [99]	Jul-2023	70	-	✓	✓	2T tokens	-	2000 80G A100	-	✓	-
	Baichuan2 [100]	Sep-2023	13	-	✓	✓	2.6T tokens	-	1024 A800	-	✓	-
	QWEN [101]	Sep-2023	14	-	✓	✓	3T tokens	-	-	-	✓	-
	FLM [102]	Sep-2023	101	-	✓	-	311B tokens	-	192 A800	22 d	✓	-
	Skywork [103]	Oct-2023	13	-	-	-	3.2T tokens	-	512 80G A800	-	✓	-
Closed Source	GPT-3 [55]	May-2020	175	-	-	-	300B tokens	-	-	-	✓	-
	GShard [104]	Jun-2020	600	-	-	-	1T tokens	-	2048 TPU v3	4 d	-	-
	Codex [105]	Jul-2021	12	GPT-3	-	-	100B tokens	May-2020	-	-	✓	-
	ERNIE 3.0 [106]	Jul-2021	10	-	-	-	375B tokens	-	384 V100	-	✓	-
	Jurassic-1 [107]	Aug-2021	178	-	-	-	300B tokens	-	800 GPU	-	✓	-
	HyperCLOVA [108]	Sep-2021	82	-	-	-	300B tokens	-	1024 A100	13.4 d	✓	-
	FLAN [67]	Sep-2021	137	LaMDA-PT	✓	-	-	-	128 TPU v3	60 h	✓	-
	Yuan 1.0 [109]	Oct-2021	245	-	-	-	180B tokens	-	2128 GPU	-	✓	-
	Anthropic [110]	Dec-2021	52	-	-	-	400B tokens	-	-	-	✓	-
	WebGPT [81]	Dec-2021	175	GPT-3	-	✓	-	-	-	-	✓	-
	Gopher [64]	Dec-2021	280	-	-	-	300B tokens	-	4096 TPU v3	920 h	✓	-
	ERNIE 3.0 Titan [111]	Dec-2021	260	-	-	-	-	-	-	-	✓	-
	GLaM [112]	Dec-2021	1200	-	-	-	280B tokens	-	1024 TPU v4	574 h	✓	-
	LaMDA [68]	Jan-2022	137	-	-	-	768B tokens	-	1024 TPU v3	57.7 d	-	-
	MT-NLG [113]	Jan-2022	530	-	-	-	270B tokens	-	4480 80G A100	-	✓	-
	AlphaCode [114]	Feb-2022	41	-	-	-	967B tokens	Jul-2021	-	-	-	-
	InstructGPT [66]	Mar-2022	175	GPT-3	✓	✓	-	-	-	-	✓	-
	Chinchilla [34]	Mar-2022	70	-	-	-	1.4T tokens	-	-	-	✓	-
	PaLM [56]	Apr-2022	540	-	-	-	780B tokens	-	6144 TPU v4	-	✓	✓
	AlexaTM [115]	Aug-2022	20	-	-	-	1.3T tokens	-	128 A100	120 d	✓	✓
	Sparrow [116]	Sep-2022	70	-	-	✓	-	-	64 TPU v3	-	✓	-
	WeLM [117]	Sep-2022	10	-	-	-	300B tokens	-	128 A100 40G	24 d	✓	-
	U-PaLM [118]	Oct-2022	540	PaLM	-	-	-	-	512 TPU v4	5 d	✓	✓
	Flan-PaLM [69]	Oct-2022	540	PaLM	✓	-	-	-	512 TPU v4	37 h	✓	✓
	Flan-U-PaLM [69]	Oct-2022	540	U-PaLM	✓	-	-	-	-	-	✓	✓
	GPT-4 [46]	Mar-2023	-	-	✓	✓	-	-	-	-	✓	✓
	PanGu-Σ [119]	Mar-2023	1085	PanGu-α	-	-	329B tokens	-	512 Ascend 910	100 d	✓	-
	PaLM2 [120]	May-2023	16	-	✓	-	100B tokens	-	-	-	✓	✓

Zhao et al.: A Survey of Large Language Models. 2024. arXiv:2303.18223

See you next week!

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Next time: Prompt engineering and efficient adaptation
  - Zero-shot, in-context learning, chain-of-thought, …
  - Prompt tuning, adapter tuning, LoRA, ….