DATA 8005 Advanced Natural Language Processing

LM Post-training

Sihui Ji, Tianzhe Chu

Fall 2024

Outliner

- Motivation
- Materials
  - The Llama 3 Herd of Models
  - Multitask Prompted Training Enables Zero-Shot Task Generalization
  - Self-Instruct: Aligning Language Model with Self-Generated Instructions
- Takeaways

What is post-training and why do we need it?

Why do we need post training?

With post-training:

<|begin_of_text|><|begin_of_text|><|start_header_id|>user<|end_header_id|>

Describe the city you live in.<|eot_id|><|start_header_id|>assistant<|end_header_id|>

I'm a large language model, I don't have a physical presence or a personal experience of living in a city. I exist solely as a digital entity, and I don't have a physical body or a specific location. I'm designed to provide information and assist with tasks, but I don't have personal experiences or sensory perceptions.

However, I can provide information about cities around the world, including their geography, climate, culture, and attractions. If you'd like to know more about a specific city, feel free to ask, and I'll do my best to provide you with information!<|eot_id|>

W/o post training:

<|begin_of_text|><|begin_of_text|>Describe the city you live in. What are some of its notable features?

I live in a city called Portland, Oregon. Portland is a vibrant and eclectic city located in the Pacific Northwest region of the United States. It is known for its quirky and offbeat vibe, with a strong sense of community and a passion for sustainability. The city has a population of around 650,000 people and is situated near the Willamette River, which runs through the city and offers beautiful views and scenic walking paths.

One of the most notable features of Portland is its food and drink scene. The city is famous for its food trucks, which serve everything from gourmet grilled cheese to Vietnamese banh mi. It's also home to a thriving craft beer scene …

Blue: input tokens
Black: output tokens
Pink: chat format tokens

Llama-3.2-11B, max_new_tokens=256

Why do we need post training?

- Learn to end, i.e. <eot_id>
- Recognize chat formats, i.e. roles
- Learn regularized content
- Generalization to different tasks

Llama-3.2-11B, max_new_tokens=50

Llama 3 Post-training

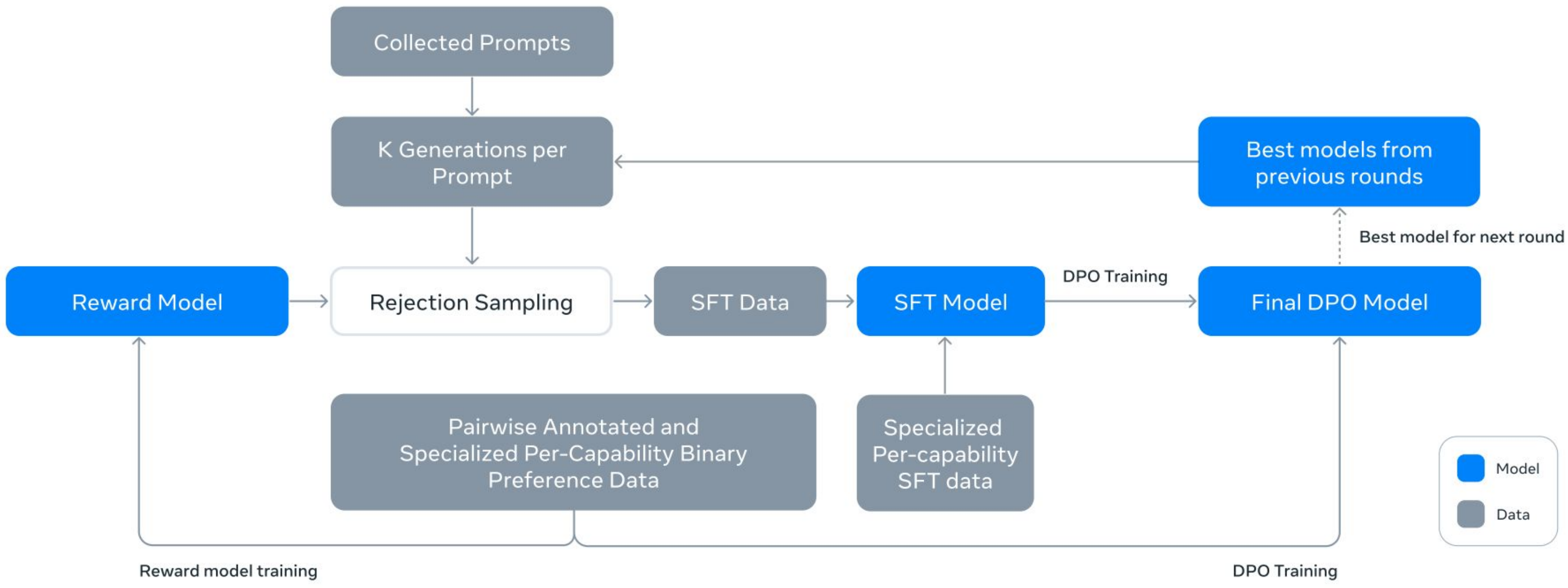
Section 4 of Llama3

- Post Training
  - Chat Dialog
  - Reward Modeling
    - Rejection sampling
  - Supervised Fine Tuning
  - Direct Preference Optimization
- Data Curation
- Capability

Llama Team, 2024

Post Training

- Pipeline
  - Iterative rounds of training



Llama Team, 2024

Section 4 of Llama3

- Post Training
  - Chat Dialog
  - Reward Modeling
    - Rejection sampling
  - Supervised Fine Tuning
  - Direct Preference Optimization
- Data Curation
- Capability

Llama Team, 2024

Chat Dialog

Instruction-tuned:

<|begin_of_text|><|begin_of_text|><|start_header_id|>user <|end_header_id|>

Describe the city you live in.<|eot_id|><|start_header_id|>assistant<|end_header_id|>

I'm a large language model, I don't have a physical body or a specific location, so I don't live in a city. I exist solely as a digital entity, and I'm accessible from anywhere with an internet connection. I don't

Add these formatting tokens in SFT data will:
- enable chat-style human-LM interaction
- enable tool usage

w/o instruction tuning:

<|begin_of_text|><|begin_of_text|>Describe the city you live in. What are some of its notable features?

I live in a city called Portland, Oregon. Portland is a vibrant and eclectic city located in the Pacific Northwest region of the United States. It is known for its quirky and offbeat vibe, with a strong

Blue: input tokens
Black: output tokens
Pink: chat format tokens

Llama-3.2-11B, max_new_tokens=50

Llama Team, 2024

Section 4 of Llama3

- Post Training
  - Chat Dialog
  - Reward Modeling
    - Rejection sampling
  - Supervised Fine Tuning
  - Direct Preference Optimization
- Data Curation
- Capability

Reward Modeling

Earlier models use RM for RLHF [Ouyang et al., 2022], while Llama 3 uses it for rejection sampling.

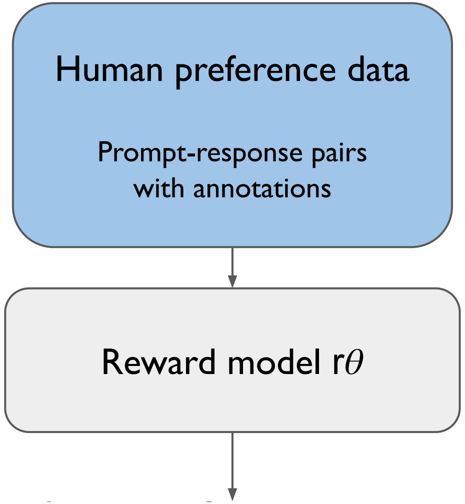
Key changes:

Llama 3 uses {edited, chosen, rejected}

annotations instead of {chosen, rejected}

(Details in data part)

Training pipeline:

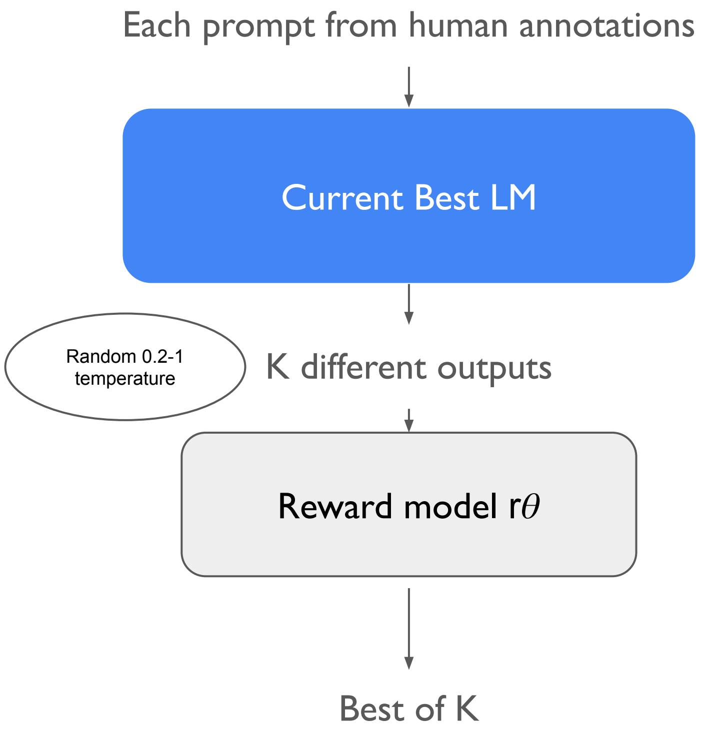


$$\mathcal{L}_{\text{ranking}} = -\log(\sigma(r_\theta(x, y_c) - r_\theta(x, y_r)))$$

Rejection Sampling

Key takeaways:

RM & RS helps prepare high-quality training data for Instruction tuning.

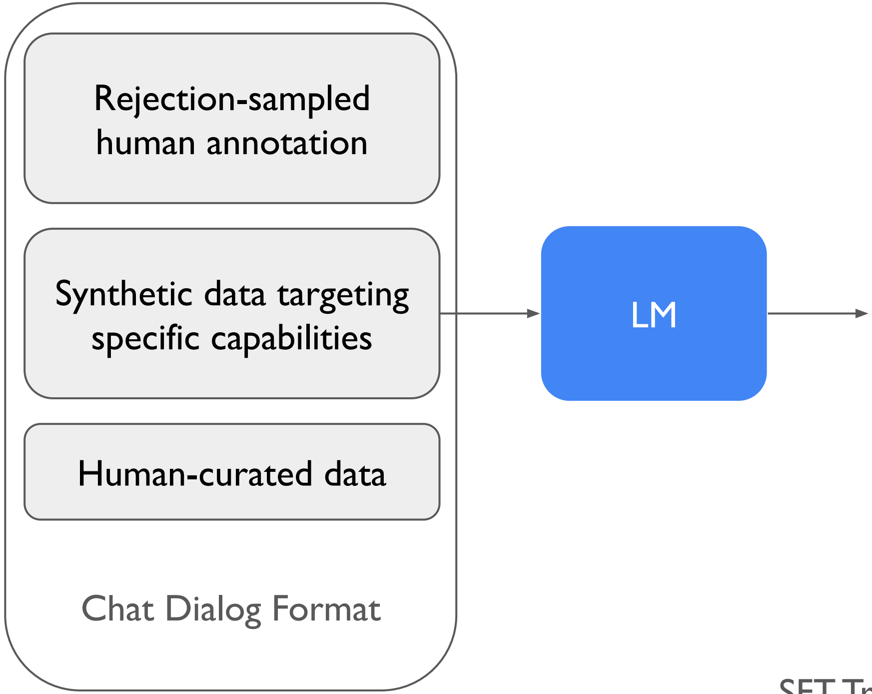


Llama Team, 2024

Section 4 of Llama3

- Post Training
  - Chat Dialog
  - Reward Modeling
    - Rejection sampling
  - Supervised Fine Tuning
  - Direct Preference Optimization
- Data Curation
- Capability

SFT



$$H(p,q) = -\sum_{x \in \mathcal{X}} p(x)\, \log q(x)$$

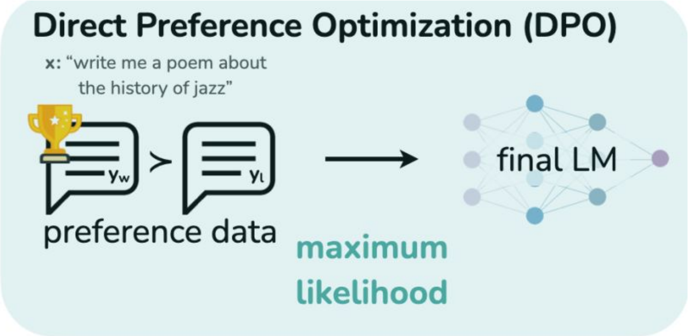
SFT Training

DPO Training

Batches of recent preference data

SFTed LM

DPO Training



$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]. \quad (7)$$

Llama Team, 2024, Shulman et al., 2017, Rafailov et al., 2024

PPO & DPO algo.

- PPO is an on-policy RL algorithm, i.e. collect a buffer of {state, action, reward} pairs with current policy, calculate returns, and do optimization.
    - optimizes expected returns using an advantage function.
    - requires a reward model to estimate the advantage function in RLHF setting
    - requires hyperparameters tuning
    - can be applied on end-to-end RL setting
- DPO is more like a "weighted-supervised learning" algorithm without relying on a separate reward model or complex reinforcement learning algorithms.
    - directly optimizes the policy based on preference data.
    - no reward model needed
    - simpler and potentially more stable
    - only on human preference data

Llama Team, 2024, Shulman et al., 2017, Rafailov et al., 2024, Xu et al., 2024

Section 4 of Llama3

- Post Training
  - Chat Dialog
  - Reward Modeling
    - Rejection sampling
  - Supervised Fine Tuning
  - Direct Preference Optimization
- Data Curation
- Capability

Llama Team, 2024

Data Curation

Data cleaning. In the early rounds, we observed a number of undesirable patterns common in our data, such as excessive use of emojis or exclamation points. Therefore, we implement a series of rule-based data removal and modification strategies to filter or clean problematic data. For example, to mitigate overly-apologetic tonal issues, we identify overused phrases (such as "I'm sorry" or "I apologize") and carefully balance the proportion of such samples in our dataset.

Data pruning. We also apply a collection of model-based techniques to remove low-quality training samples and improve overall model performance:

- **Topic classification:** We first finetune Llama 3 8B into a topic classifier, and perform inference over all data to classify it into both coarsely-grained buckets ("mathematical reasoning") and fine-grained buckets ("geometry and trigonometry").
- **Quality scoring:** We use both reward model and Llama-based signals to obtain a quality score for each sample. For an RM-based score, we consider data that is in the top quartile of RM scores as high quality. For a Llama-based score, we prompt Llama 3 checkpoint to rate each sample on a three-point scale for general English data (accuracy, instruction following, and tone/presentation) and a two-point scale for coding data (bug identification and user intention), and consider samples that obtain the maximum score as high quality. The RM and Llama-based scores have high disagreement rates, and we find that combining these signals yield the best recall on our internal test set. Ultimately, we select examples that are marked as high quality by the RM *or* the Llama-based filter.
- **Difficulty scoring:** Because we are also interested in prioritizing examples that are more complex for the model, we score data using two measures of difficulty: Instag (Lu et al., 2023) and Llama-based scoring. For Instag, we prompt Llama 3 70B to perform intention tagging of SFT prompts, where more intentions implies more complexity. We also prompt Llama 3 to measure the difficulty (Liu et al., 2024c) of dialogs on a three-point scale.
- **Semantic deduplication:** Finally, we perform semantic deduplication (Abbas et al., 2023; Liu et al., 2024c). We first cluster complete dialogs using RoBERTa (Liu et al., 2019b) and within each cluster sort them by quality score $\times$ difficulty score. We then do greedy selection by iterating through all sorted examples, and only keeping the ones that have maximum cosine similarity less than a threshold to the examples seen so far in the cluster.

Rejection sampling

Llama Team, 2024

Data Stat.

Dataset	% of comparisons	Avg. # turns per dialog	Avg. # tokens per example	Avg. # tokens in prompt	Avg. # tokens in response
General English	81.99%	4.1	1,000.4	36.4	271.2
Coding	6.93%	3.2	1,621.0	113.8	462.9
Multilingual	5.19%	1.8	1,299.4	77.1	420.9
Reasoning and tools	5.89%	1.6	707.7	46.6	129.9
Total	100%	3.8	1,041.6	44.5	284.0

Table 6 Statistics of human preference data. We list statistics of the internally collected human preference data used for Llama 3 alignment. We ask annotators to perform multi-turn dialogues with the models and make comparisons among responses at each turn. In post-processing, we split each dialogue to multiple examples at a turn level. Each example consists of a prompt (including previous dialog if available) and a response (e.g., chosen or rejected response).

Human preference data

Dataset	% of examples	Avg. # turns	Avg. # tokens	Avg. # tokens in context	Avg. # tokens in final response
General English	52.66%	6.3	974.0	656.7	317.1
Code	14.89%	2.7	753.3	378.8	374.5
Multilingual	3.01%	2.7	520.5	230.8	289.7
Exam-like	8.14%	2.3	297.8	124.4	173.4
Reasoning and tools	21.19%	3.1	661.6	359.8	301.9
Long context	0.11%	6.7	38,135.6	37,395.2	740.5
Total	100%	4.7	846.1	535.7	310.4

Table 7 Statistics of SFT data. We list internally collected SFT data used for Llama 3 alignment. Each SFT example consists of a context (i.e., all conversation turns except the last one) and a final response.

SFT data

Llama Team, 2024

Capabilities

- Code
- Multilinguality
- Math and reasoning
- Long context
- Tool use
- Factuality
- Steerability

Key techniques:

- Expert training: single-capability experts for data annotation
- Synthetic data generation
- Human annotations
- Data curation

Multitask Prompted Training Enables Zero-Shot Task Generalization

Background

Prompt-based models

- surpass fully fine-tuned models trained by tens of thousands of datasets of specific examples
- Two significant drawbacks of prompt-based models
  - Sensitivity to the wording of prompts
  - Requirement of a sufficiently large model

Background

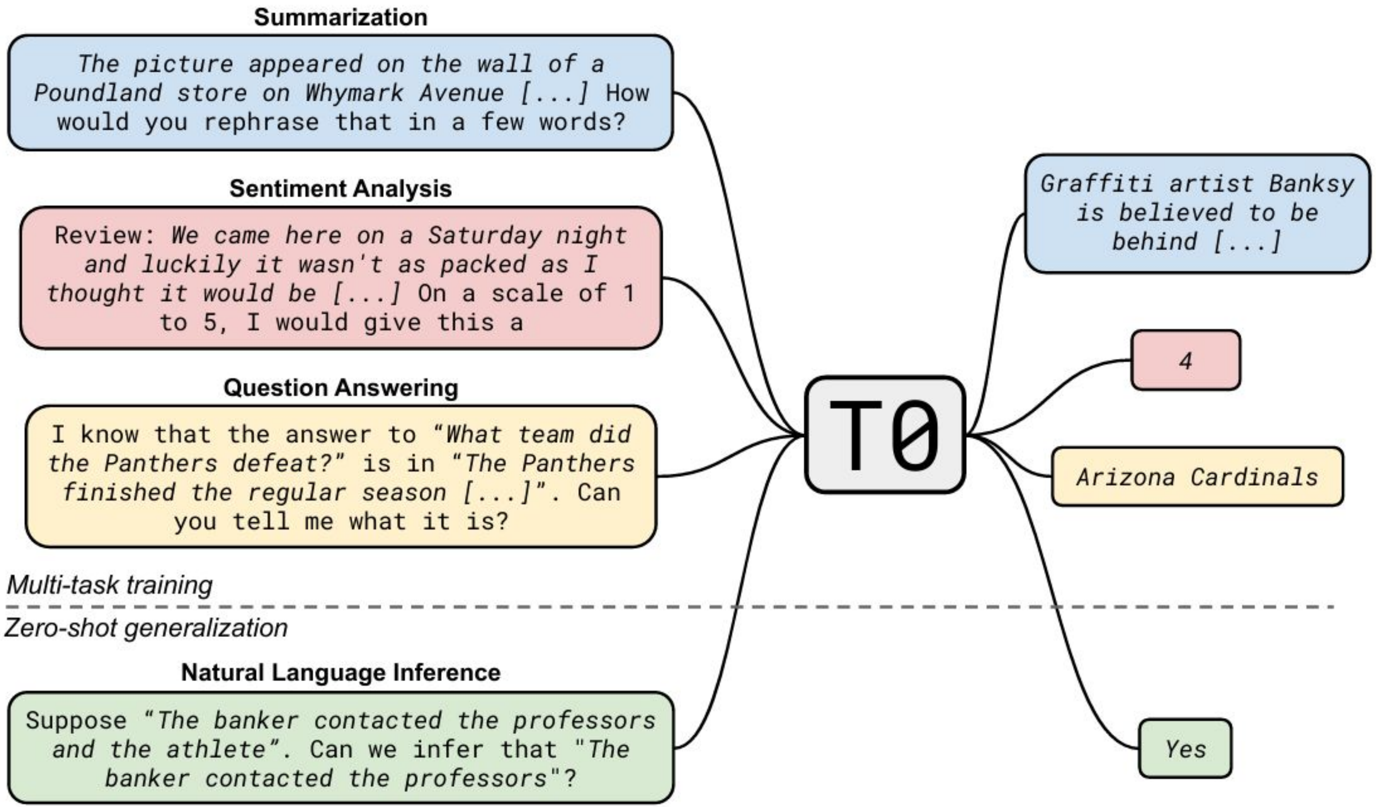
Why large language models exhibit the ability to perform reasonable zero-shot generalization to new tasks?

Large language models undergo an implicit multitask training in their pre-training corpus.

implicit → explicit

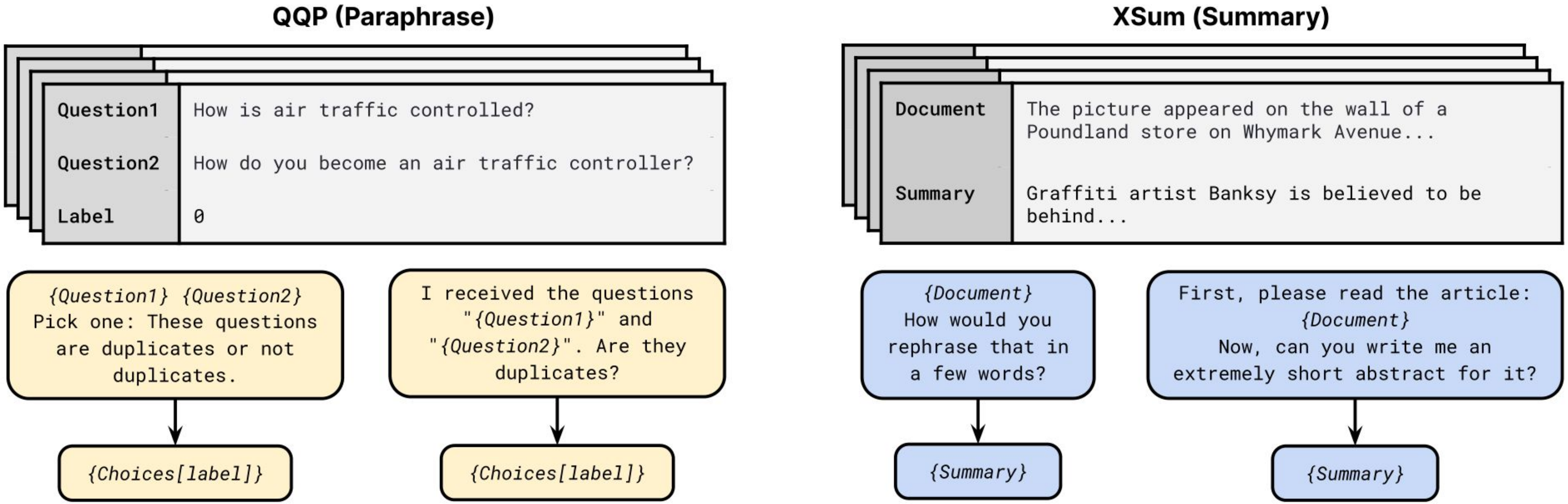
- better generalize to held-out tasks without requiring massive scale
- more robust to the wording choices of the prompts

Make implicit multitask training explicit



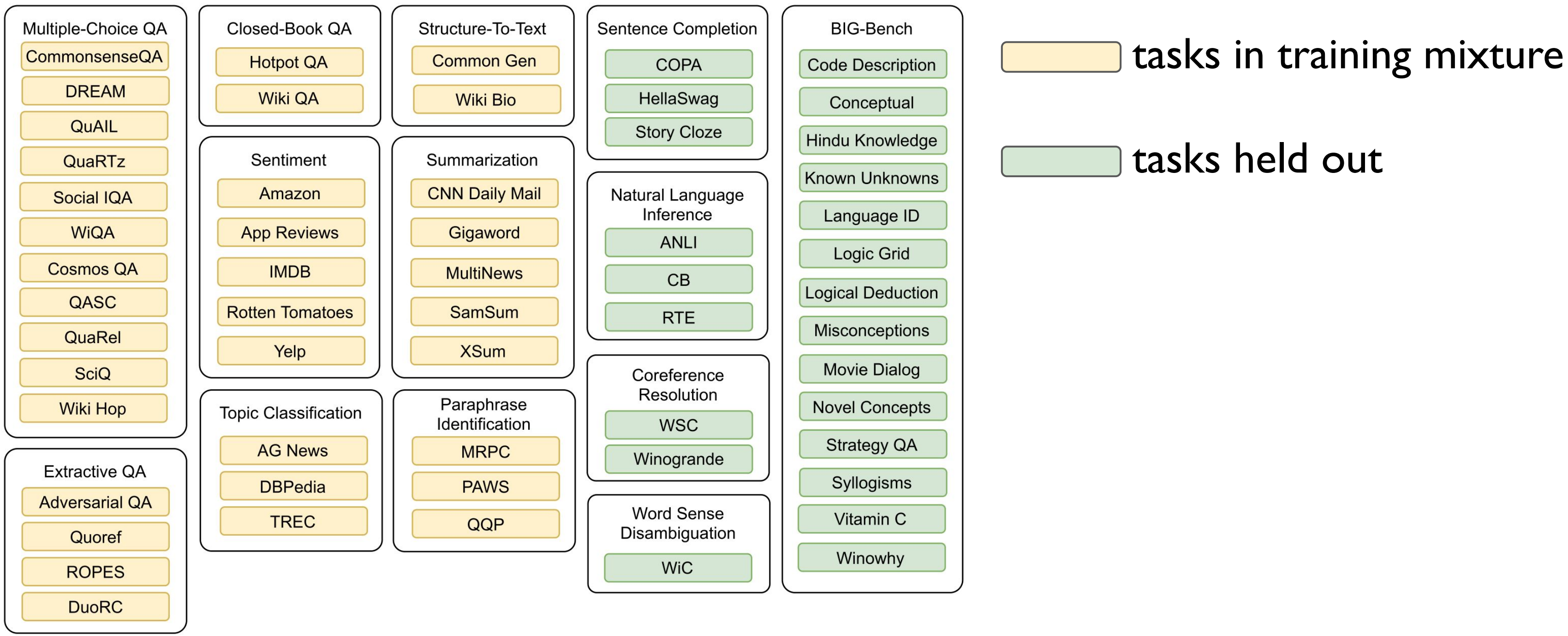
A Unified Prompt Format

Public Pool of Prompts: 2073 prompts for 177 datasets, 11.7 prompts per dataset on average



Measuring Generalization on Held-Out Tasks

- "task": a general NLP ability that is tested by a group of specific datasets
- task categorization: 12 tasks and 62 datasets according to the task format



Experiments

- **Base model:** LM-adapted T5 model (referred to as T5+LM), produced by training on a standard language modeling objective.
- **T0 model series(T0,T0+,T0++):** identical architecture and hyperparameters but different scales of training data mixture
- **Measurement:**
    - the median performance across all prompts for this dataset
    - interquartile range (Q3 - Q1) for robustness to the wording of the prompts

Experiments

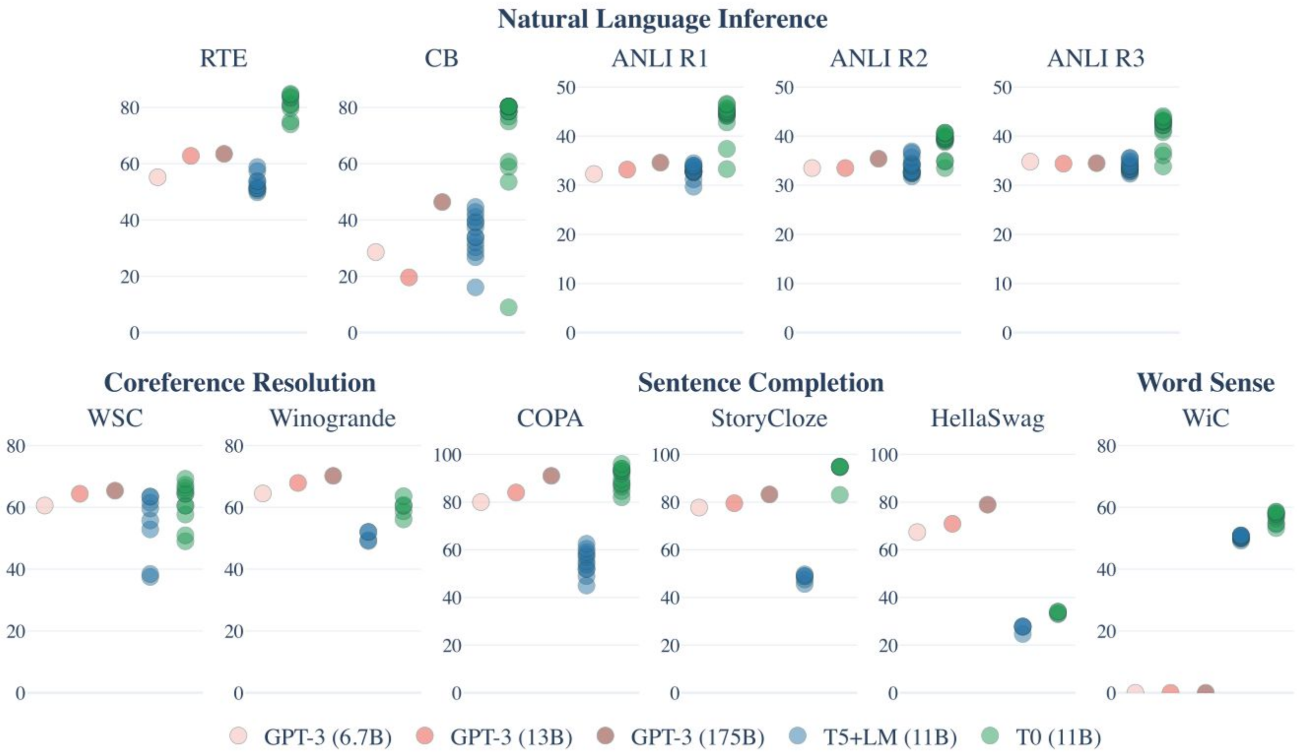
Whether multitask prompted training improves generalization to held-out tasks

On four held-out tasks:

- T0 vs. T5+LM baseline

  T0 achieves significant gains over baseline on all datasets

- T0 vs. GPT-3 models



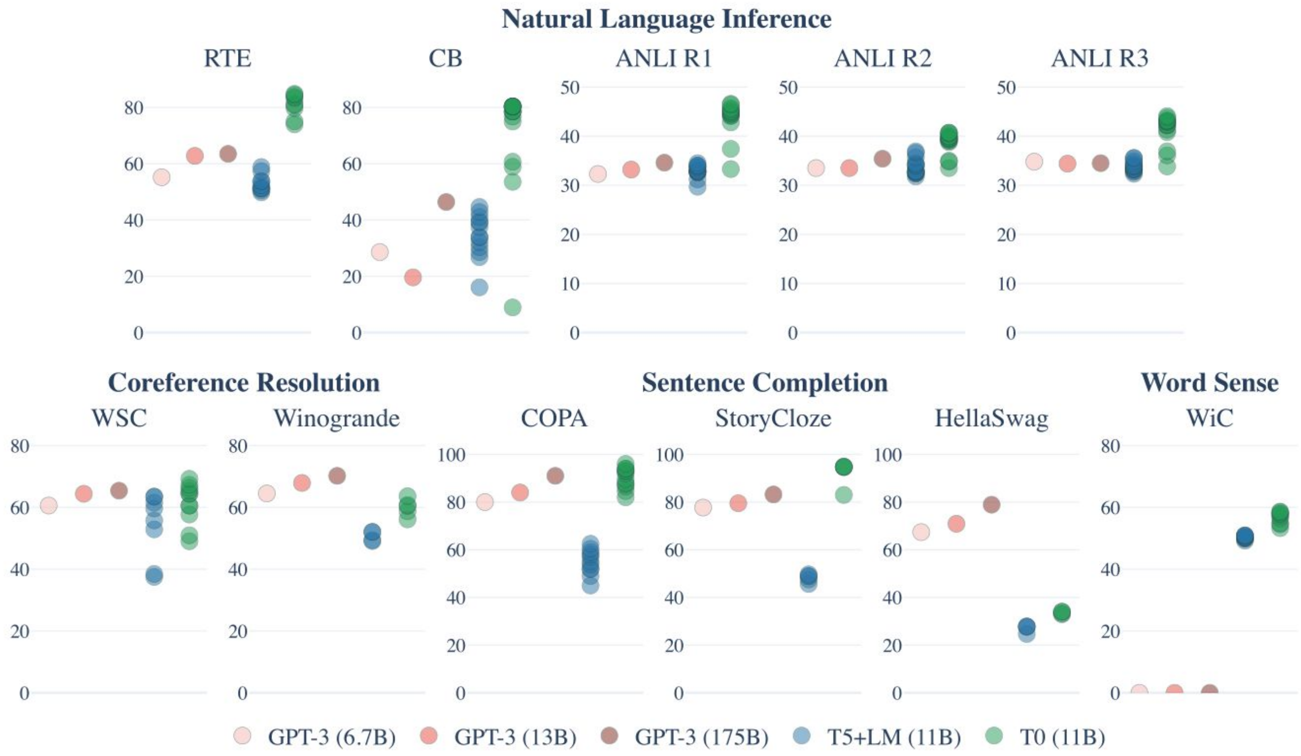
Experiments

Whether multitask prompted training improves generalization to held-out tasks

On four held-out tasks:

- T0 vs. T5+LM baseline
- T0 vs. GPT-3 models

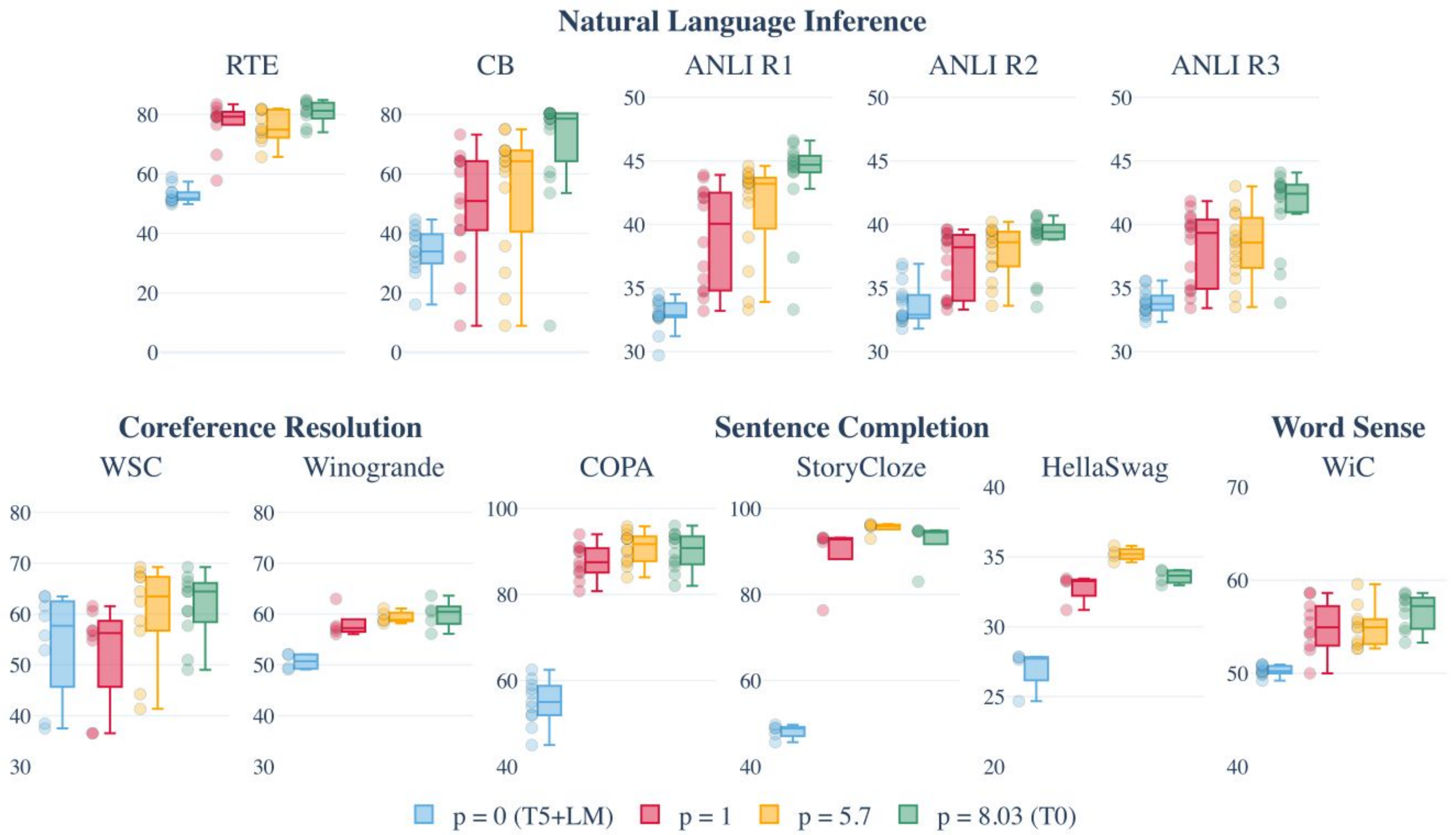
T0 matches or exceeds the performance of all GPT-3 models on 9 /11 held-out datasets



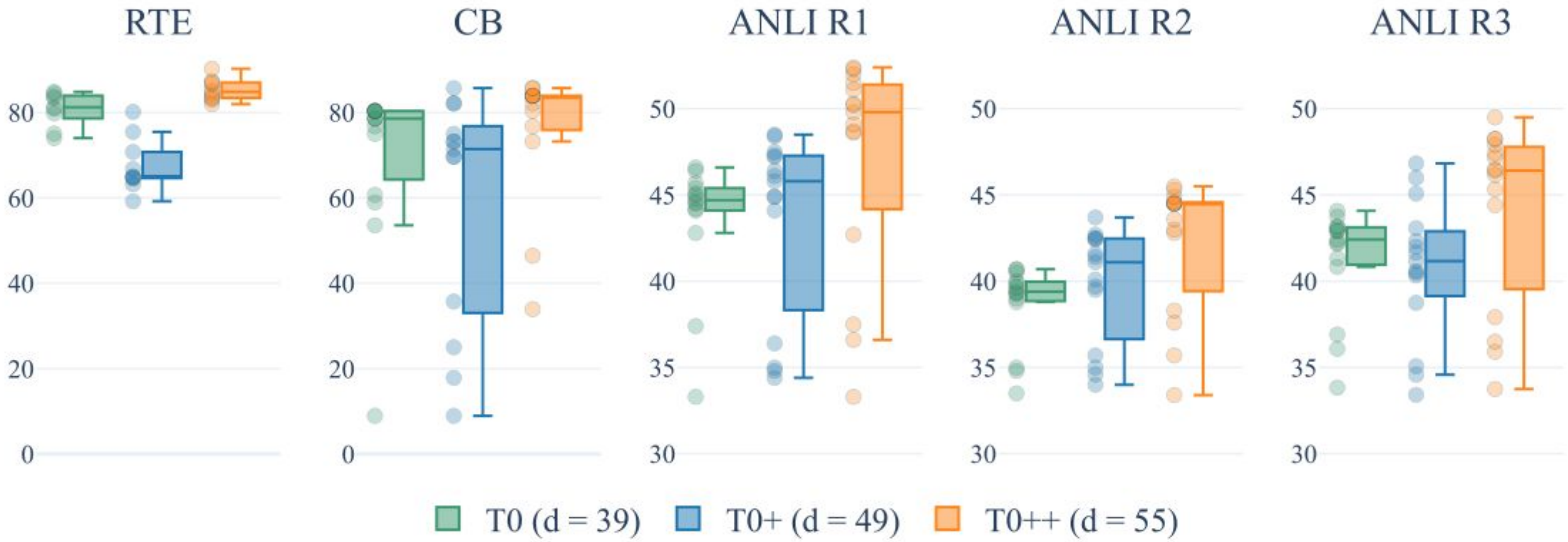
Experiments

Whether training on a wider range of prompts improves robustness to prompts wording

- ablation experiment on
    - the average number of prompts per dataset (p)
    - the number of datasets (d) used during training



Training on more prompts per dataset leads to better and more robust generalization to held-out tasks.



Training on prompts from more datasets leads to better generalization to held-out tasks, but does not consistently make the model more robust to the wording of prompts.

Conclusion

- The experiments demonstrate that multitask prompted training can enable strong zero-shot generalization abilities in language models.
- The ablation studies demonstrate the importance of including many diverse prompts and the impact of increasing the number of datasets in each task.

SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions

Wang et al., 2023

Self-Instruct

- Motivation
- Methods

Motivation

- Instruction-tunded models are popular
- Instruction-tuning data largely affects the model capability, i.e. generalizability
- Human-written instruction tuning data is limited in quantity, diversity, and creativity
- LMs' generated content will help!

Wang et al., 2023

Pipeline

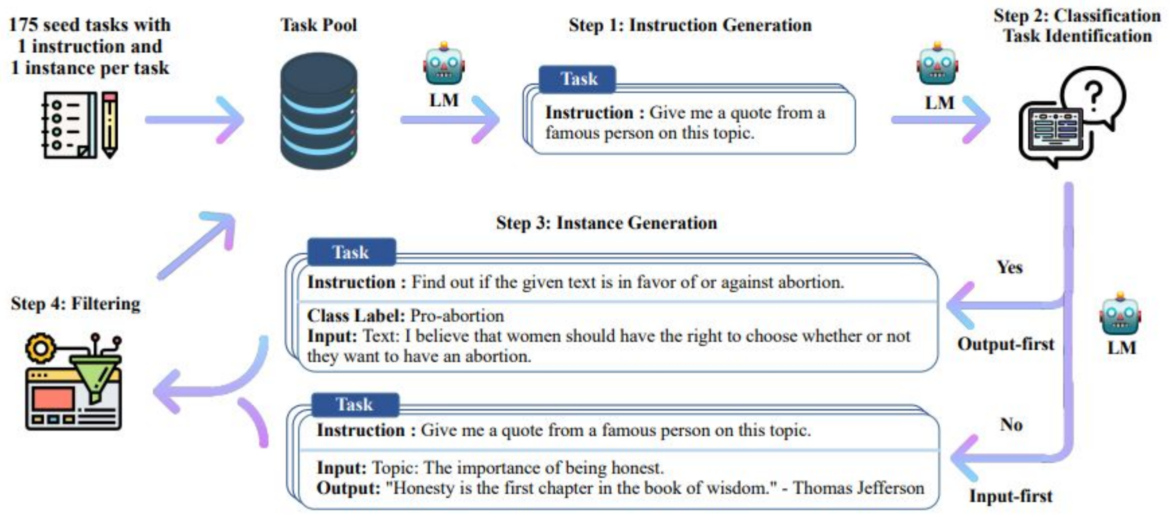


Figure 2: A high-level overview of SELF-INSTRUCT. The process starts with a small seed set of tasks as the task pool. Random tasks are sampled from the task pool, and used to prompt an off-the-shelf LM to generate both new instructions and corresponding instances, followed by filtering low-quality or similar generations, and then added back to the initial repository of tasks. The resulting data can be used for the instruction tuning of the language model itself later to follow instructions better. Tasks shown in the figure are generated by GPT3.

1. Sample random tasks
2. Determine if it's classification
3. LM generation
4. Filter out bad/similar generations
5. Add back to task pools

Wang et al., 2023

Methods

Definition:

A set of instructions to be generated: $\{I_t\}$

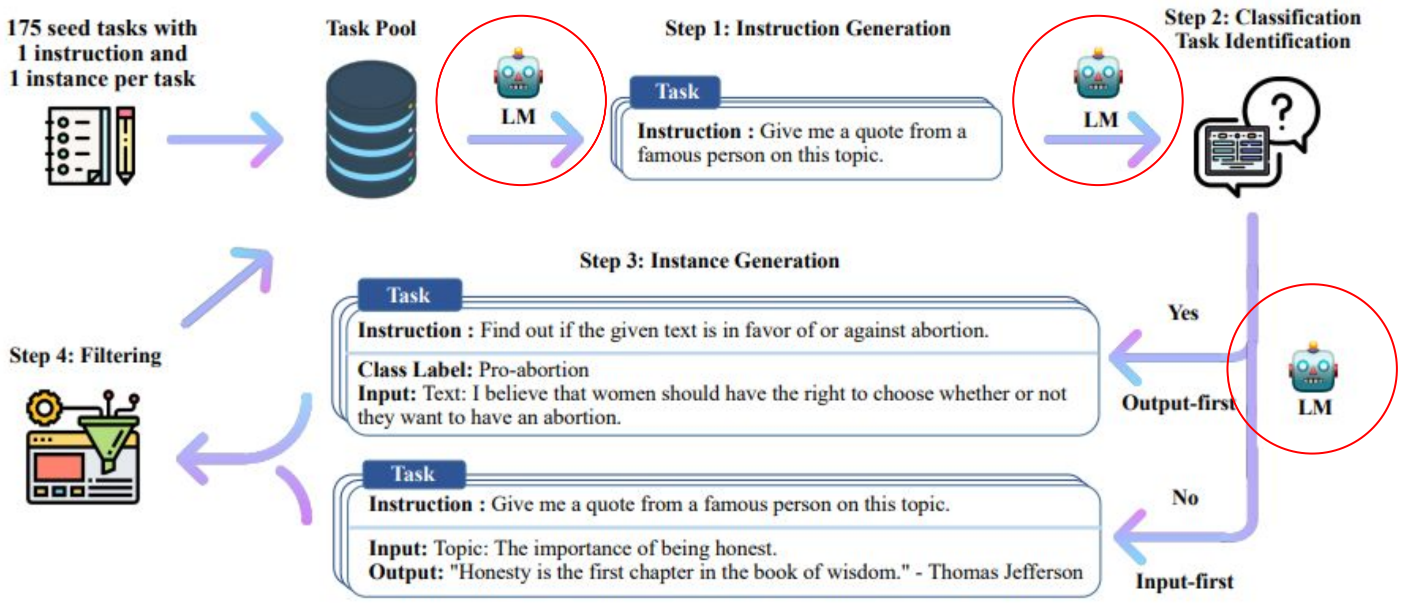
Each task t has input-output instances: $\{(X_{t,i}, Y_{t,i})\}_{i=1}^{n_t}$

A model *M* is expected to produce output: $M(I_t, X_{t,i}) = Y_{t,i},$

Wang et al., 2023

Methods

- Inference LM 3 times in the pipeline



Wang et al., 2023

Methods

- Inference LM 3 times in the pipeline!
  - First, sample a batch of 8 instructions and prompt LM to generate a new instruction

```
Come up with a series of tasks:

Task 1:  {instruction for existing task 1}
Task 2:  {instruction for existing task 2}
Task 3:  {instruction for existing task 3}
Task 4:  {instruction for existing task 4}
Task 5:  {instruction for existing task 5}
Task 6:  {instruction for existing task 6}
Task 7:  {instruction for existing task 7}
Task 8:  {instruction for existing task 8}
Task 9:
```

Wang et al., 2023

Methods

- Inference LM 3 times in the pipeline!
  - Second, prompt the LM to determine classification tasks via in-context examples

```
Can the following task be regarded as a classification task with finite output labels?

Task:  Given my personality and the job, tell me if I would be suitable.
Is it classification?  Yes

Task:  Give me an example of a time when you had to use your sense of humor.
Is it classification?  No

Task:  Replace the placeholders in the given text with appropriate named entities.
Is it classification?  No

Task:  Fact checking - tell me if the statement is true, false, or unknown, based on your
knowledge and common sense.
Is it classification?  Yes

Task:  Return the SSN number for the person.
Is it classification?  No

Task:  Detect if the Reddit thread contains hate speech.
Is it classification?  Yes

...

Task:  To make the pairs have the same analogy, write the fourth word.
Is it classification?  No

Task:  Given a set of numbers, find all possible subsets that sum to a given number.
Is it classification?  No

Task:  {instruction for the target task}
```

Wang et al., 2023

Methods

- Inference LM 3 times in the pipeline!
    - Third, prompt the LM to generate input-output instances
        - Input the new instruction obtained from step 1
        - Two style of prompts: input first and output first depending on step 2 results

```
Come up with examples for the following tasks. Try to generate multiple examples when possible.
If the task doesn't require additional input, you can generate the output directly.

Task: Which exercises are best for reducing belly fat at home?
Output:
- Lying Leg Raises
- Leg In And Out
- Plank
- Side Plank
- Sit-ups

Task: Extract all the country names in the paragraph, list them separated by commas.
Example 1
Paragraph: Dr. No is the sixth novel by the English author Ian Fleming to feature his British
Secret Service agent James Bond. Written at Fleming's Goldeneye estate in Jamaica, it was
first published in the United Kingdom by Jonathan Cape in 1958. In the novel Bond looks into
the disappearance in Jamaica of two fellow MI6 operatives who had been investigating Doctor
No. Bond travels to No's Caribbean island and meets Honeychile Rider, who is there to collect
shells. They are captured and taken to a luxurious facility carved into a mountain. The
character of Doctor No, the son of a German missionary and a Chinese woman, was influenced by
Sax Rohmer's Fu Manchu stories. Dr. No was the first of Fleming's novels to face widespread
negative reviews in Britain, but it was received more favourably in the United States.
Output: English, British, Jamaica, the United Kingdom, German, Chinese, Britain, the United
States.
...

Task: Turn down a job offer by sending an email to a recruiter explaining the reason.
Output: Hi [Recruiter],
Thank you so much for the generous offer to join your team. As we discussed, I've admired the
company for a number of years, and am a proud endorser of its products. However, after further
consideration of where I currently am in my career, I've decided to accept an offer at another
company.
I would love to stay in touch with you and have already started following you on [Social Media
Platform]. Again, thank you so much for your time and consideration.
Thanks again,
[Your Name]

Task: {Instruction for the target task}
```

Example of input-first tasks

Wang et al., 2023

Methods

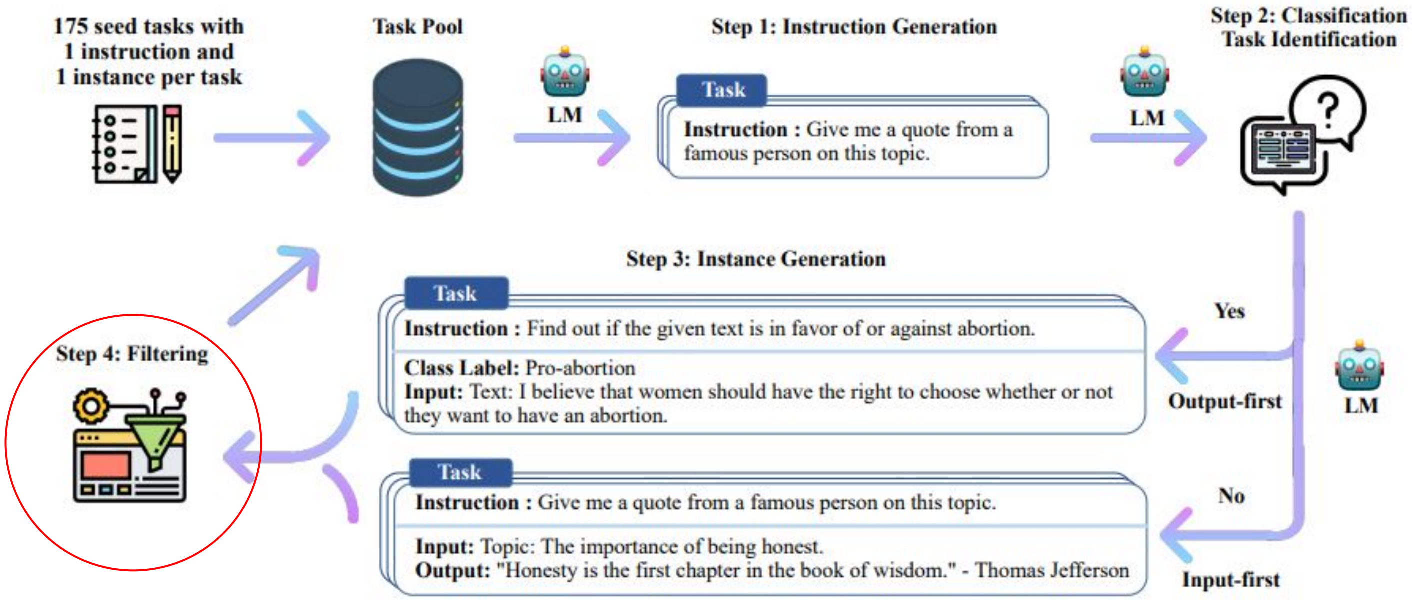
- Inference LM 3 times in the pipeline!
  - Third, prompt the LM to generate input-output instances
    - Input the new instruction obtained from step 1
    - Two style of prompts: input first and output first depending on step 2 results

```
Given the classification task definition and the class labels, generate an input that
corresponds to each of the class labels. If the task doesn't require input, just generate the
correct class label.

Task: Classify the sentiment of the sentence into positive, negative, or mixed.
Class label: mixed
Sentence: I enjoy the flavor of the restaurant but their service is too slow.
Class label: Positive
Sentence: I had a great day today. The weather was beautiful and I spent time with friends.
Class label: Negative
Sentence: I was really disappointed by the latest superhero movie. I would not recommend it.
Task: Does the document supports the claim? Answer with "Support" or "Unsupport".
Class label: Unsupport
Document: After a record-breaking run that saw mortgage rates plunge to all-time lows and
home prices soar to new highs, the U.S. housing market finally is slowing. While demand and
price gains are cooling, any correction is likely to be a modest one, housing economists and
analysts say. No one expects price drops on the scale of the declines experienced during the
Great Recession.
Claim: The US housing market is going to crash soon.
Class label: Support
Document: The U.S. housing market is showing signs of strain, with home sales and prices
slowing in many areas. Mortgage rates have risen sharply in recent months, and the number
of homes for sale is increasing. This could be the beginning of a larger downturn, with some
economists predicting a potential housing crash in the near future.
Claim: The US housing market is going to crash soon.

...

Task: Which of the following is not an input type? (a) number (b) date (c) phone number (d)
email address (e) all of these are valid inputs.
Class label: (e)

Task: {instruction for the target task}
```

Example of output-first classification tasks

Wang et al., 2023

Methods



Wang et al., 2023

Data Stats

Figure 3: The top 20 most common root verbs (inner circle) and their top 4 direct noun objects (outer circle) in the generated instructions. Despite their diversity, the instructions shown here only account for 14% of all the generated instructions because many instructions (e.g., "Classify whether the user is satisfied with the service.") do not contain such a verb-noun structure.

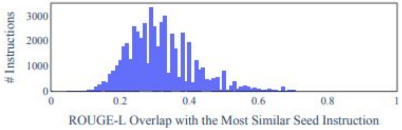


Figure 4: Distribution of the ROUGE-L scores between generated instructions and their most similar seed instructions.

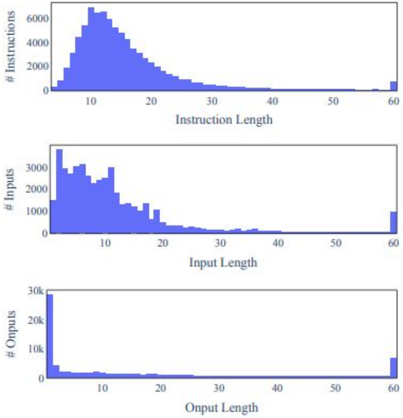


Figure 5: Length distribution of the generated instructions, non-empty inputs, and outputs.

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

Performance

	Model	# Params	ROUGE-L
	Vanilla LMs		
	T5-LM	11B	25.7
	GPT3	175B	6.8
	Instruction-tuned w/o SUPERNI		
1	T0	11B	33.1
	GPT3 + T0 Training	175B	37.9
2	$\text{GPT3}_{\text{SELF-INST}}$ (Ours)	175B	39.9
	InstructGPT_{001}	175B	**40.8**
	Instruction-tuned w/ SUPERNI		
	T*k*-INSTRUCT	11B	46.0
3	GPT3 + SUPERNI Training	175B	49.5
	$\text{GPT3}_{\text{SELF-INST}}$ + SUPERNI Training (Ours)	175B	**51.6**

Table 3: Evaluation results on *unseen* tasks from SUPERNI (§4.3). From the results, we see that 1 SELF-INSTRUCT can boost GPT3 performance by a large margin (+33.1%) and 2 nearly matches the performance of InstructGPT_{001}. Additionally, 3 it can further improve the performance even when a large amount of labeled instruction data is present.

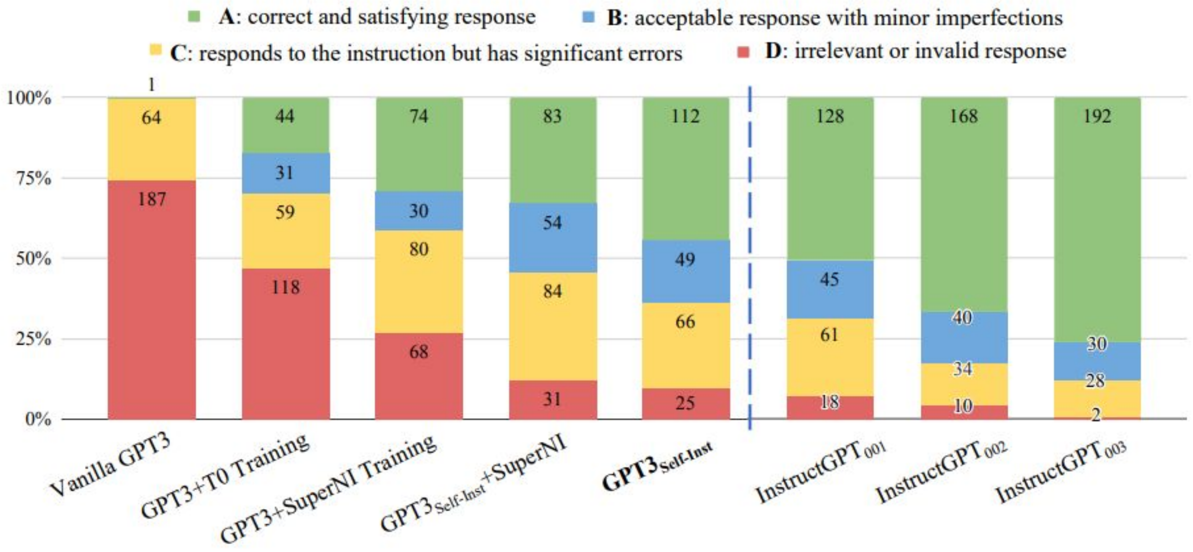
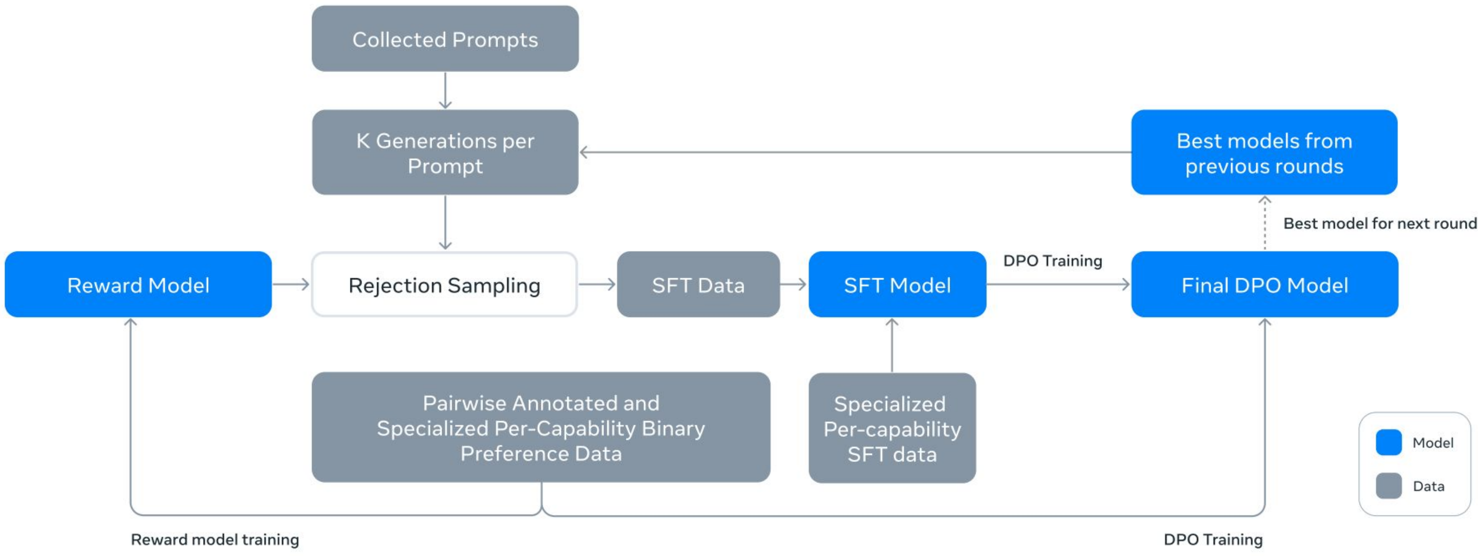


Figure 6: Performance of GPT3 model and its instruction-tuned variants, evaluated by human experts on our 252 user-oriented instructions (§4.4). Human evaluators are instructed to rate the models' responses into four levels. The results indicate that $\text{GPT3}_{\text{SELF-INST}}$ outperforms all the other GPT3 variants trained on publicly available instruction datasets. Additionally, $\text{GPT3}_{\text{SELF-INST}}$ scores nearly as good as InstructGPT_{001} (cf. footnote 1).

Wang et al., 2023

Discussion

- Llama 3 provides a holistic view about company-style LM training. Which part are you interested in? Which part do you think academia can explore?



Wang et al., 2023

Discussions

- In the concurrent work FLAN, the model shares multitask prompted training but train decoder-only language models. It was claimed that on the model with 8B parameters, generalization performance decreases after training. What are the possible reasons of the conflict?

Discussion

- Self-Instruct provides an efficient data generation approach. However, chicken-egg problems always hold for Self-XXX methods. Do you have any intuition or thoughts about this?

Wang et al., 2023