



Washington
University in St. Louis

JAMES MCKELVEY
SCHOOL OF ENGINEERING

CSE 561A: Large Language Models

Fall 2024

Lecture 4: Instruction Tuning

Jiaxin Huang

Course Announcements

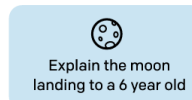
- The sign-up sheet is updated:
https://docs.google.com/spreadsheets/d/1xSCaIOjiri17V7IjP2dikFwbOgInPb_azBfZKeTgmc/edit
- Since we have 2-3 students presenting for each class, we decide to let students collectively preparing for **one** 60-min presentation including all 4 papers. Please distribute the work among yourselves, and also prepare some slides on the **background/related works**, as well as **future directions and challenges**
- The presentation score will be given based on the whole group's performance

Large Language Model Pre-training Framework

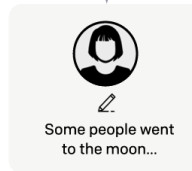
- ChatGPT training procedure
 - Self-supervised pre-training
 - Supervised training on pairs of human-written data (Step 1)
 - Model generate multiple outputs for a prompt, train a reward model on human-labeled ranking list (Step 2)
 - Optimize the language model with the trained reward model (Step 3)

Step 1
**Collect demonstration data,
and train a supervised policy.**

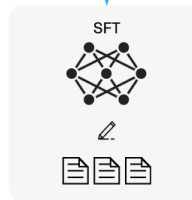
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



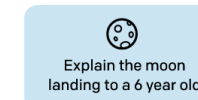
This data is used to fine-tune GPT-3 with supervised learning.



Instruction-Tuning
(covered in this course)

Step 2
**Collect comparison data,
and train a reward model.**

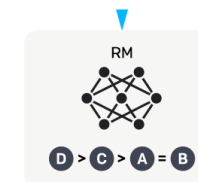
A prompt and several model outputs are sampled.



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.

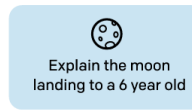


Large Language Model Pre-training Framework

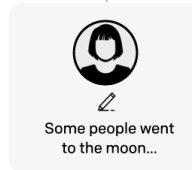
- ChatGPT training procedure
 - Model generate multiple outputs for a prompt, train a reward model on human-labeled ranking list (Step 2)
 - Optimize the language model with the trained reward model (Step 3)

Step 1
Collect demonstration data, and train a supervised policy.

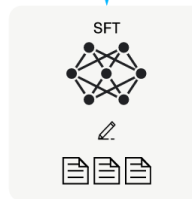
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.

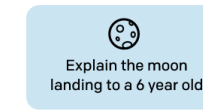


This data is used to fine-tune GPT-3 with supervised learning.



Step 2
Collect comparison data, and train a reward model.

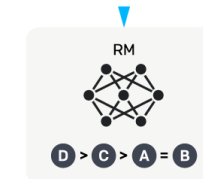
A prompt and several model outputs are sampled.



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.

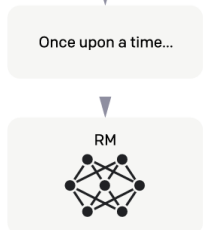
A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Reinforcement Learning from Human Feedback (covered in future courses, ~Week 5)

LLaMA: Open and Efficient Foundation Language Models

- ChatGPT (175B): **closed-source/proprietary models** where we don't know about their pre-training corpus, and don't have access to their pre-trained weights, and can only do inference or training through APIs
- LLaMA (7B/13B/33B/65B): **open-source models** where the pre-trained corpus is transparent and we have full access to their pre-trained weights
 - Pre-trained Corpus: English CommonCrawl, C4, Github, Wikipedia, Gutenberg and Books3, ArXiv, Stack Exchange.
 - Smaller models allow researchers with limited computing resources to understand how and why these language models work

Content

- **Instruction Tuning Overview**
- Instruction Tuning on Public NLP Datasets
- Instruction Tuning on Crowdsourced Datasets
- Instruction Tuning on LM-Generated Datasets
- Instruction Tuning on Mixture of Datasets

Language Models and User intents

- Language models pre-trained on large corpus not necessarily aligned with user intents

⚡ Inference API ⓘ

📄 Text2Text Generation

Who is Donald Trump?

Compute

⌘+Enter

0.7

Computation time on cpu: 0.578 s

's President? Who is Hillary Clinton? Who is Donald Trump?? Who is

⚡ Inference API ⓘ

📄 Text2Text Generation

Examples ▾

Who is Donald Trump?

Compute

⌘+Enter

0.2

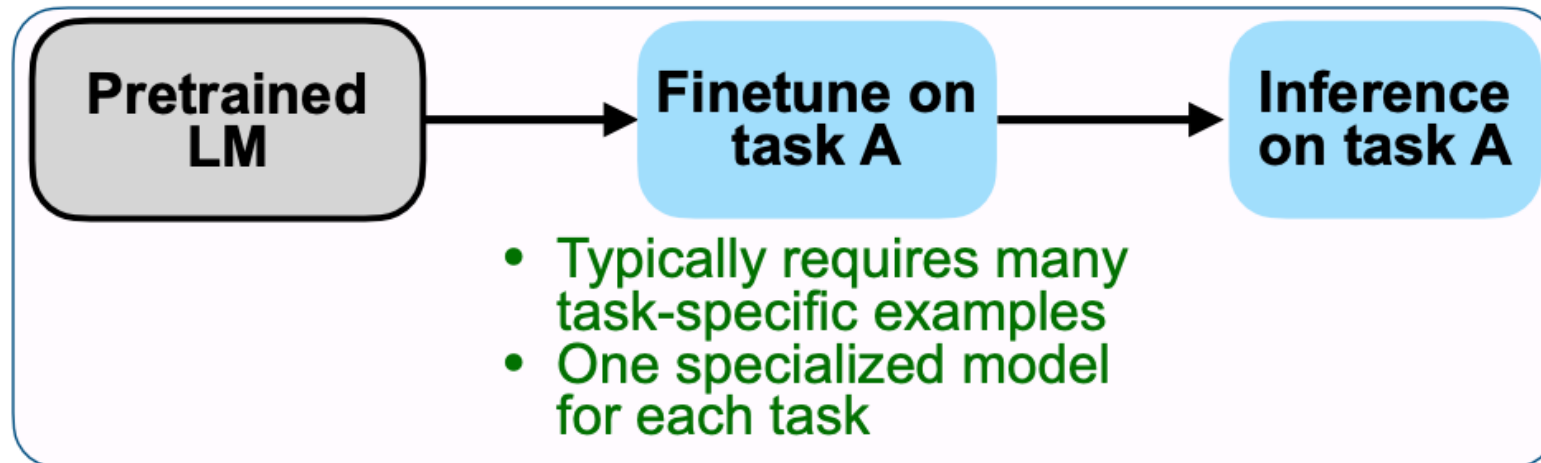
Computation time on cpu: 0.159 s

president of united states

Inference with T5 model and instruction-tuned T5 model

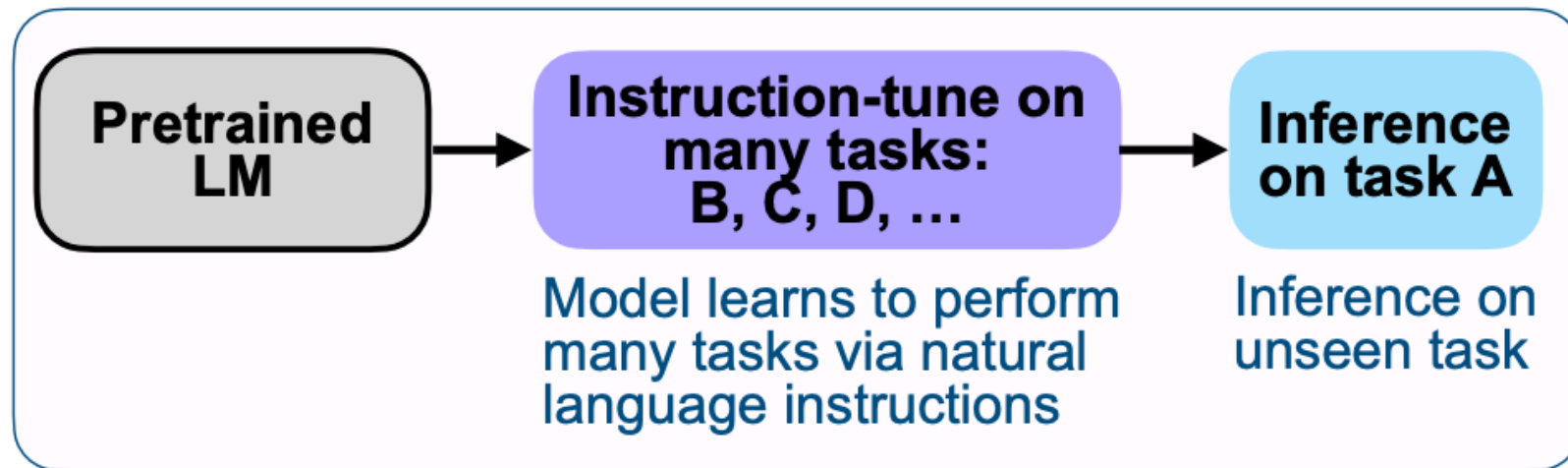
Recall Finetuning

- The pre-training stage let language models learn generic representation and knowledge from the corpus, but they are not specifically fine-tuned on any form of user tasks.
- To adapt language models to a specific downstream task, we could use task-specific datasets for fine-tuning

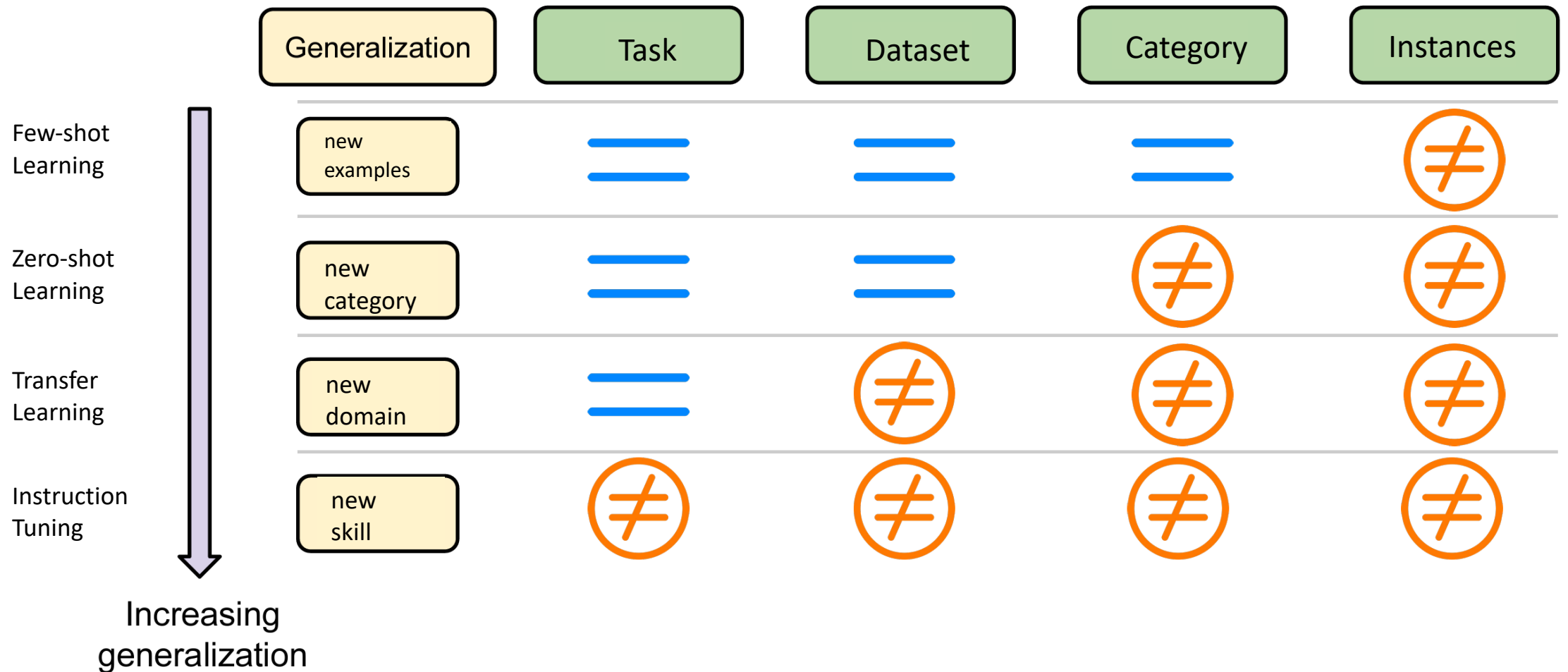


Instruction Tuning

- Fine-tuning on many tasks! Teach language models to follow different natural language instructions, so that it could perform better on downstream tasks and generalize to unseen tasks!
- Fine-tuning -> Instruction Pre-training



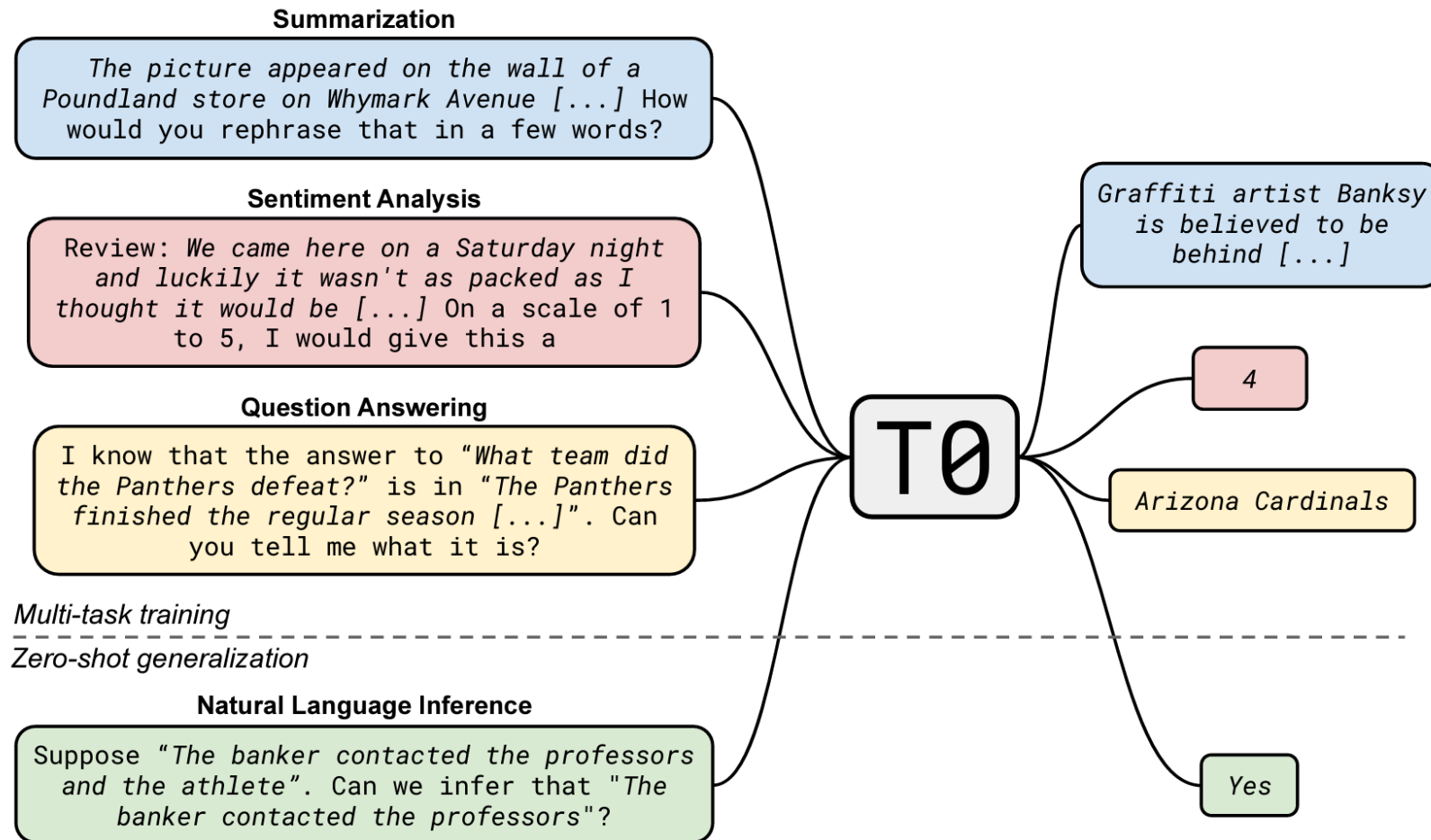
Increasing Generalization



Content

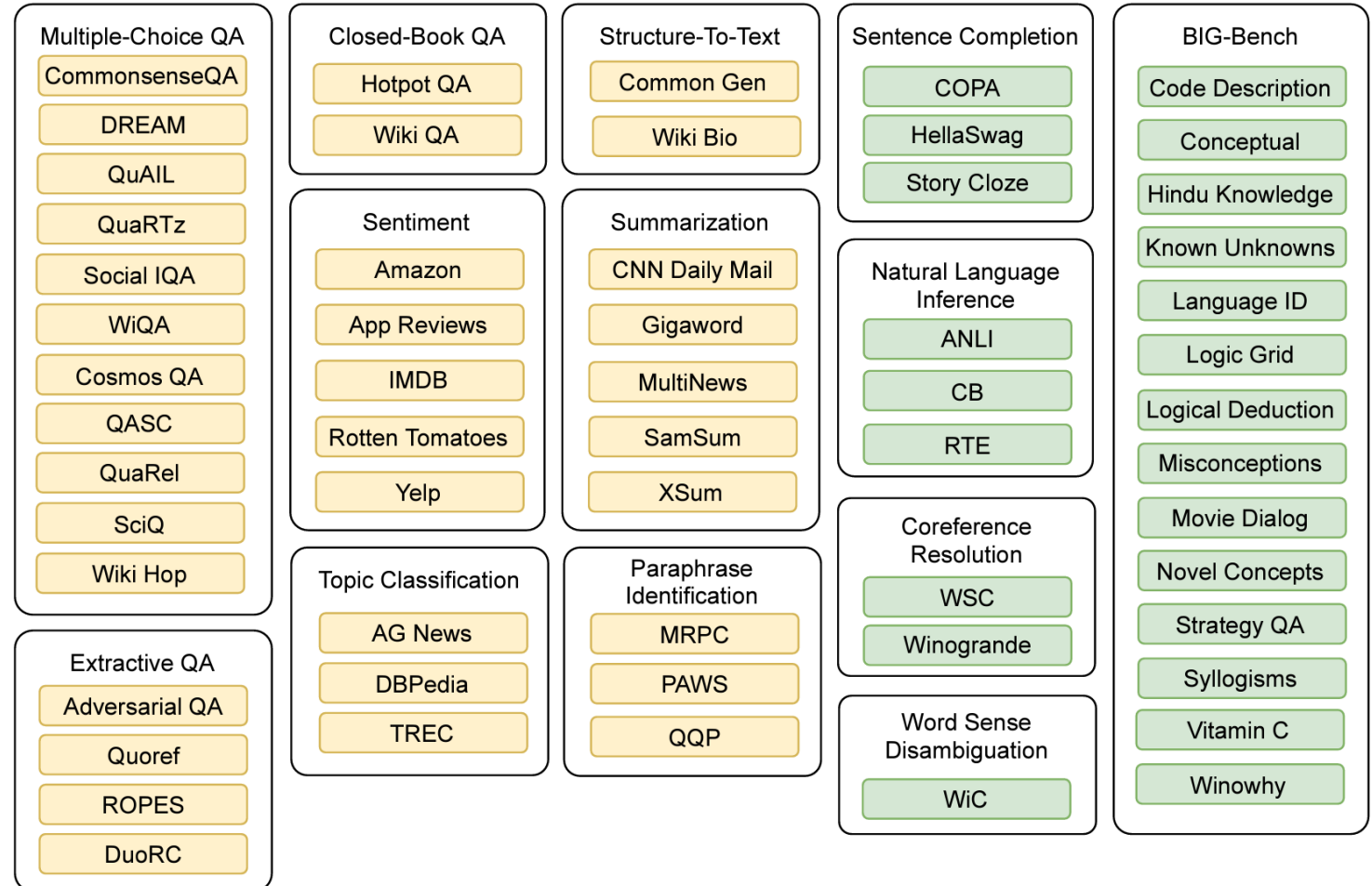
- Instruction Tuning Overview
- **Instruction Tuning on Public NLP Datasets**
- Instruction Tuning on Crowdsourced Datasets
- Instruction Tuning on LM-Generated Datasets
- Instruction Tuning on Mixture of Datasets

Multitask Prompted Training Enables Zero-Shot Task Generalization (Sanh et. al, 2021)



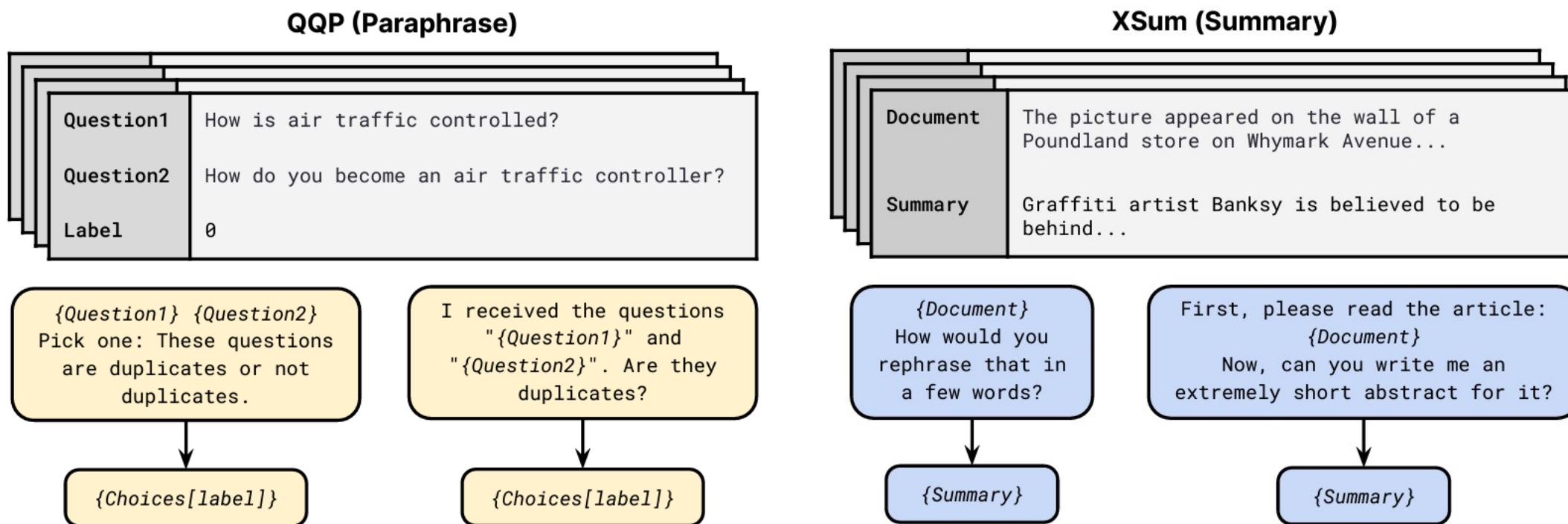
T0 Training Datasets

- Collecting from multiple public NLP datasets
- Training mixtures:
 - QA (Question Answering tasks), structure-to-text, summarization
 - Sentiment analysis, topic classification, paraphrase identification
- Held-out test set:
 - Sentence completion, BIG-Bench
 - Natural language inference, coreference resolution, word sense disambiguation



Task Adaptation with Prompt Templates

- Instead of directly using pairs of input and output, add specific instructions to explain each task (different templates per task)
- the outputs are tokens instead of class labels

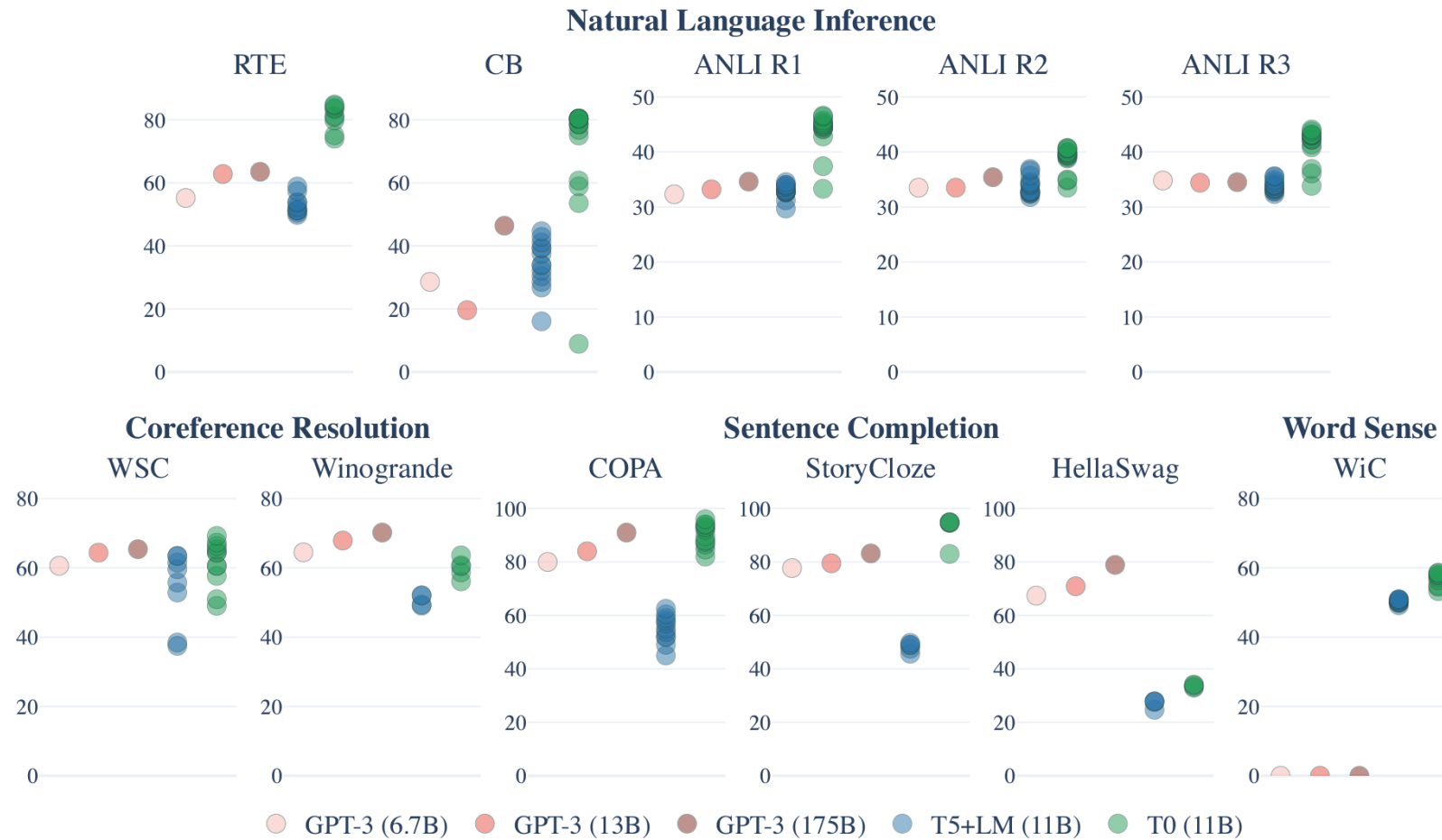


Experiments

- The multi-task trained model is called T0
- trained from T5-LM (11B model) with multitask mixture of training sets
- Baselines

● GPT-3 (6.7B) ● GPT-3 (13B) ● GPT-3 (175B) ● T5+LM (11B)

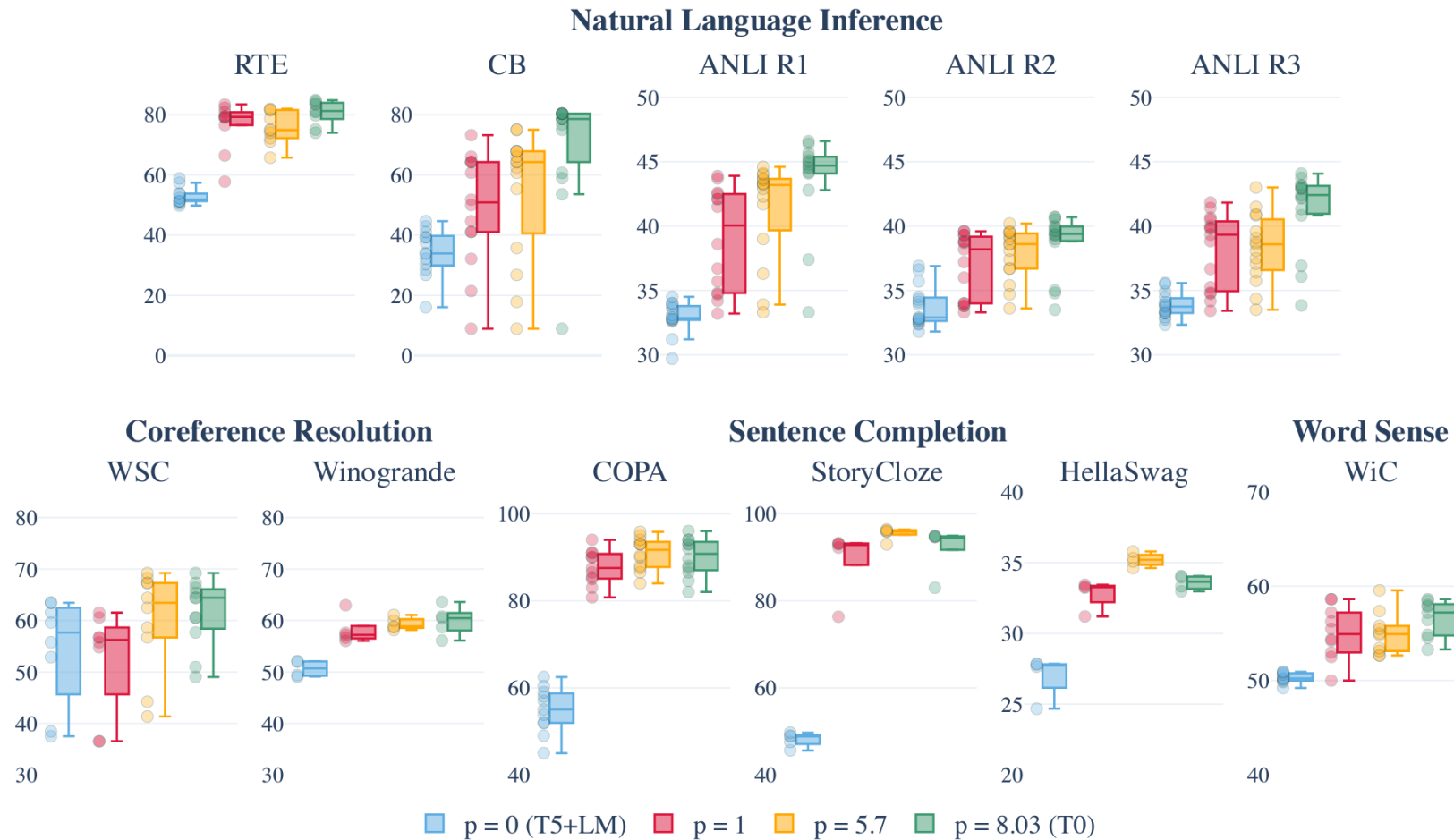
Performance on Unseen Tasks



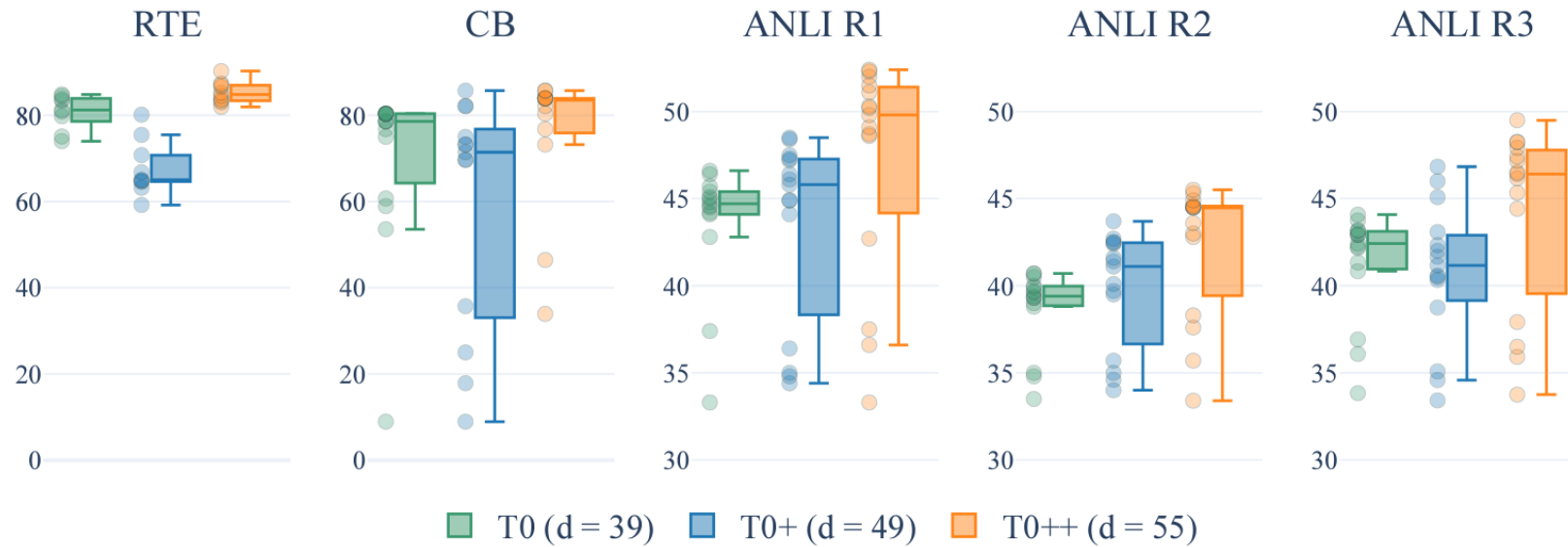
- For T5 and T0 models, each dot represents one evaluation prompt.

Effects of Prompt Numbers

- Increasing number of paraphrasing prompts for each task in training



Effects of More Training Datasets



- Adding more datasets consistently leads to higher median performance

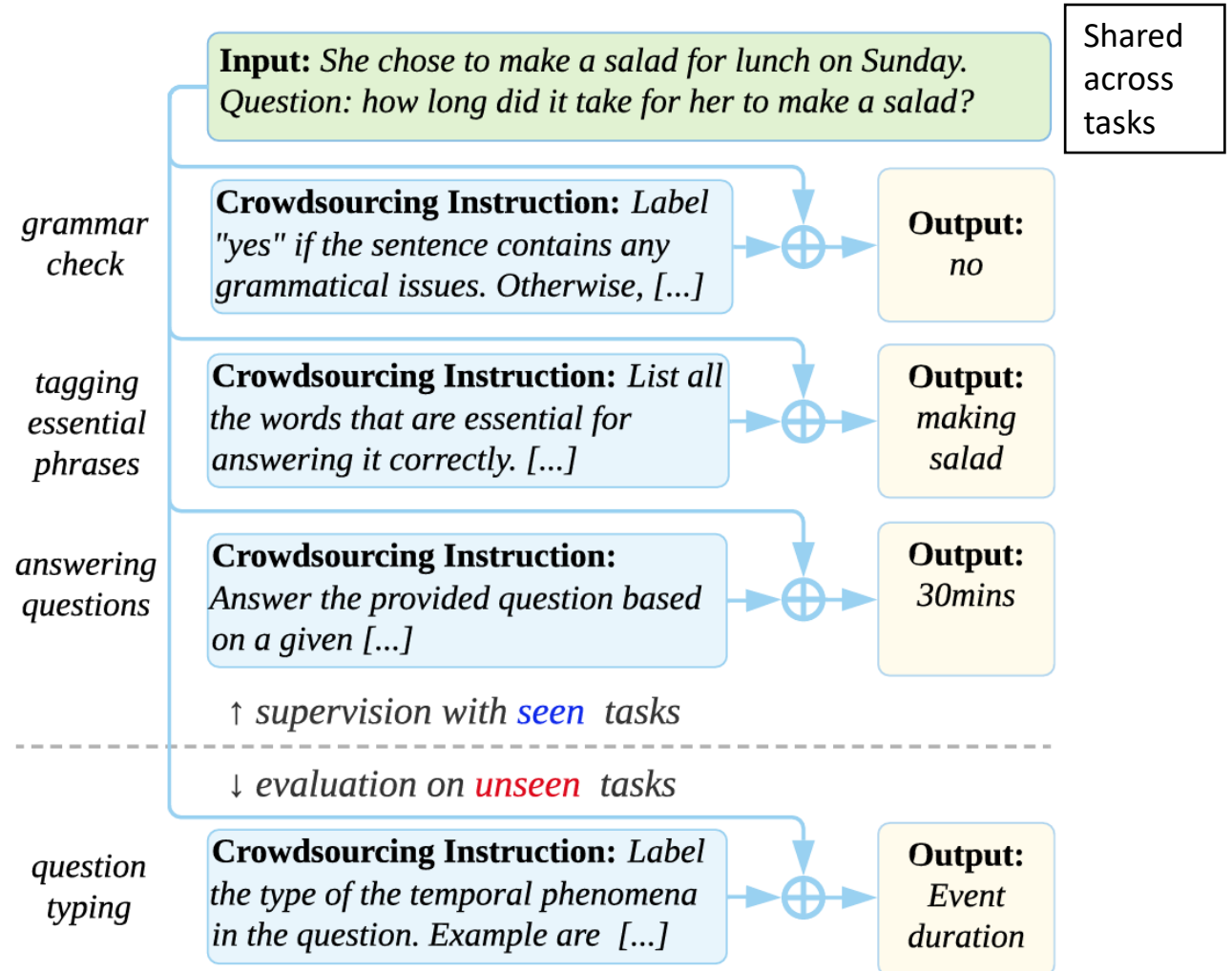
Content

- Instruction Tuning Overview
- Instruction Tuning on Public NLP Datasets
- **Instruction Tuning on Crowdsourced Datasets**
- Instruction Tuning on LM-Generated Datasets
- Instruction Tuning on Mixture of Datasets

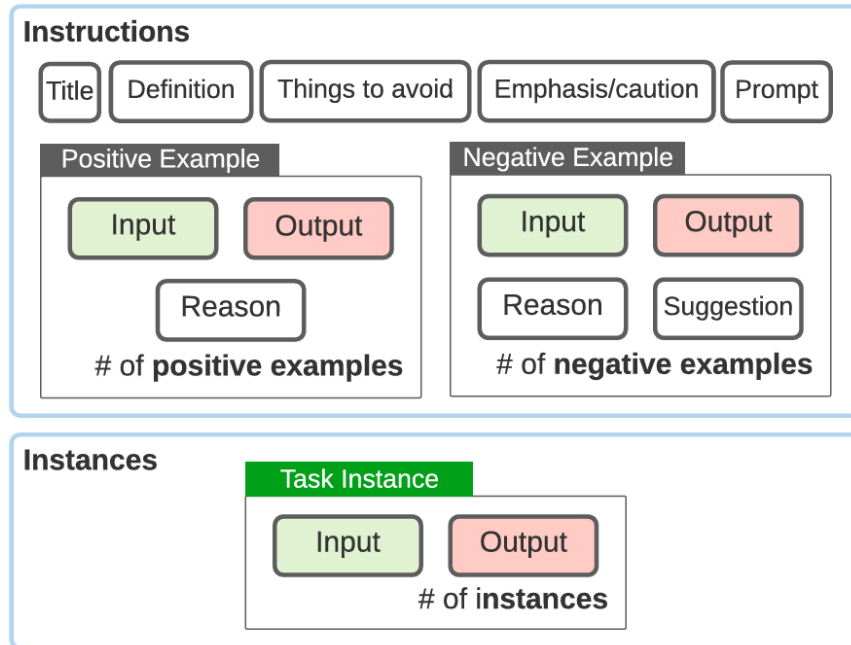
Cross-Task Generalization via Natural Language Crowdsourcing Instructions (Mishra et. al, 2021)

- Observation: conventional supervised models learned on individual datasets struggle with generalization across tasks
- A crowdsourced dataset: Natural Instructions
 - human-authored instructions
 - 61 distinct tasks
 - 193k instances (input -> output)
- A more complete instruction schema

<https://arxiv.org/abs/2104.08773>



Proposed Data Schema



- “Title” provides a high-level description of a task.
- “Definition” provides the core detailed instructions for a task.
- “Things to avoid” contain instructions regarding undesirable annotations that must be avoided.
- “Emphasis/caution” highlights statements to be emphasized or warned against.
- “Positive examples” an example of desired input/output pair.
- “Negative examples” an example of undesired input/output pair.

An Example from the Dataset

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?

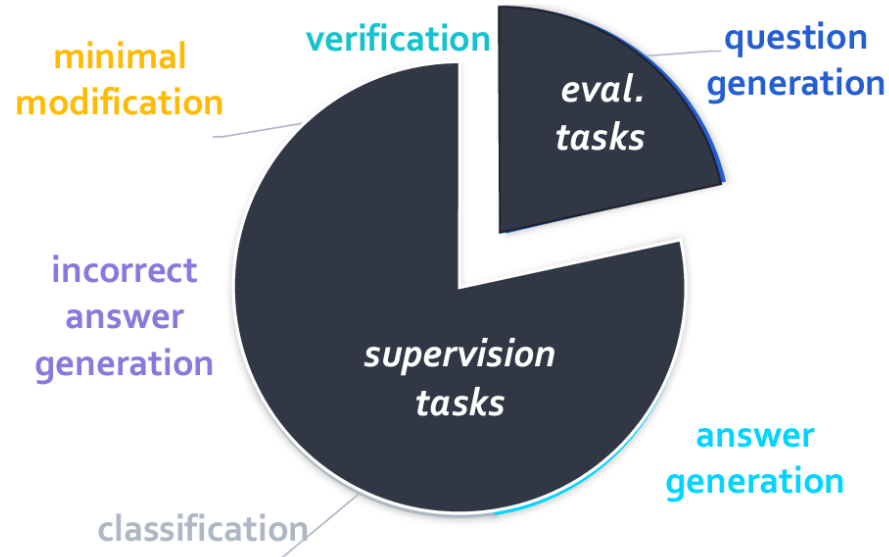
⋮

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?

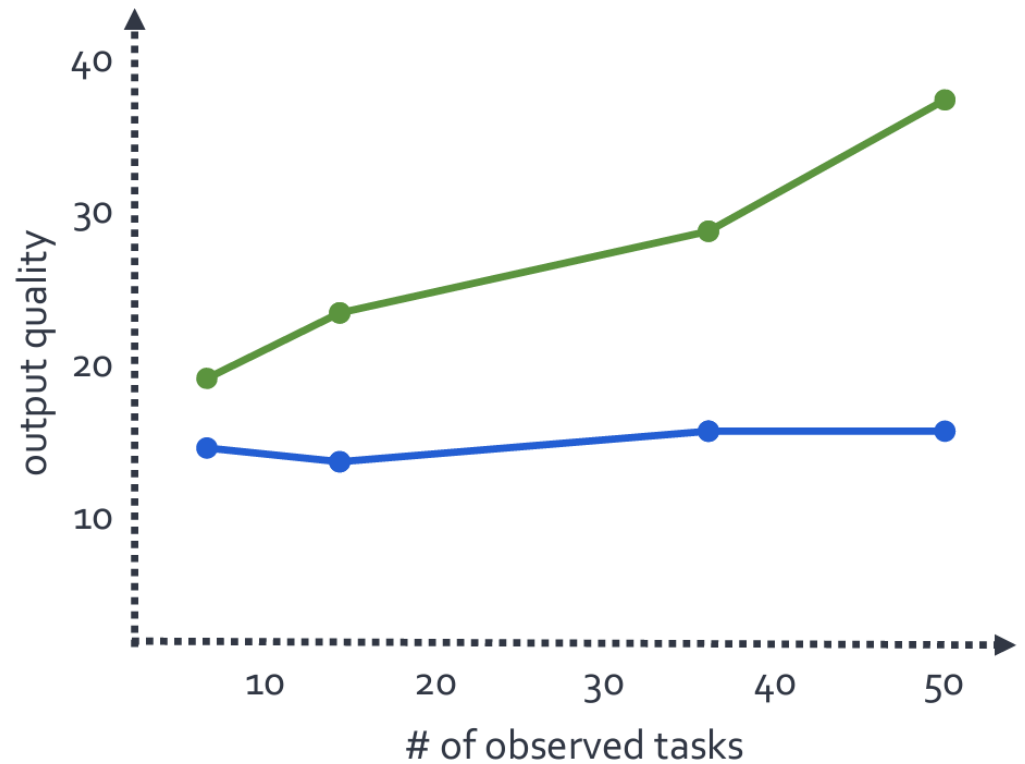
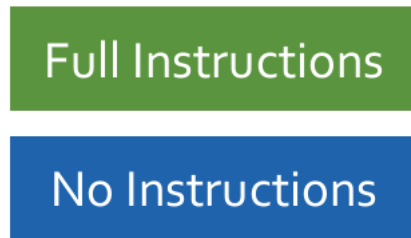
Crowdsourced Dataset

- 1. Randomly split the tasks (12 evaluation tasks, 49 supervision tasks)
- 2. Leave-one-category-out



Experiments: Number of Training Tasks

- Generalization to unseen tasks improves with more seen tasks



Experiments on the Data Schema

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

model ↓	task category →	QG	AG	CF	IAG	MM	VF	avg
BART (fine-tuned)	NO INSTRUCTION	26	6	0	21	33	7	13
	PROMPT	27	22	7	22	34	9	20
	+DEFINITION	35	24	50	25	36	7	30↑ (+50)
	+THINGS TO AVOID	33	24	4	24	58	9	25↑ (+25)
	+EMPHASIS	38	23	16	26	49	3	26↑ (+30)
	+POS. EXAMPLES	53	22	14	25	17	7	23↑ (+15)
	+DEFINITION+POS. EXAMPLES	51	23	56	25	37	6	33↑ (+65)
	POS. EXAMP.	55	6	18	25	8	6	20
FULL INSTRUCTION	46	25	52	25	35	7	32↑ (+60)	
GPT3 (not fine-tuned)	FULL INSTRUCTION	33	18	8	12	60	11	24 (+11)

QG: Question Generation, AG: Answer Generation, CF: Classification, IAG: Incorrect Answer Generation, MM: Minimal Text Modification, VF: Verification

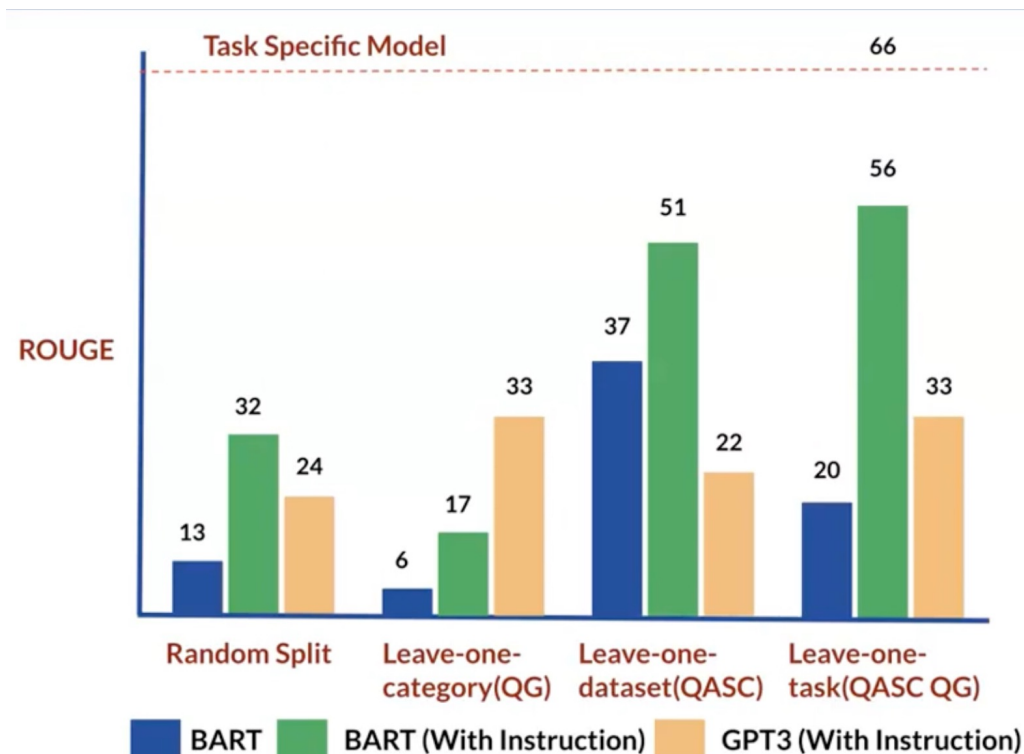
Negative Examples

Model ↓	Split ↓	w/ neg. examples	w/o neg. examples
BART	random	32	35
	leave-one- x		
	↳ x = category (AG)	19	21
	↳ x = dataset (Quoref)	37	37
	↳ x = task (QASC QG)	56	57
GPT3	-	24	44

- Negative examples can harm the result.

Experiments: Comparison with Larger Model

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



- Instructions consistently improve model performance on unseen tasks.
- When both having access to instructions on the held-out set, BART (instruction-tuned) can often outperform GPT3 (not instruction-tuned)

Content

- Instruction Tuning Overview
- Instruction Tuning on Public NLP Datasets
- Instruction Tuning on Crowdsourced Datasets
- **Instruction Tuning on LM-Generated Datasets**
- Instruction Tuning on Mixture of Datasets

Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et. al, 2022)

- Human-written instruction data can be very expensive!
- Can we reduce the human annotations?
- Idea: bootstrap from off-the-shelf LMs

Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et. al, 2022)

- Human written seed tasks to bootstrap off-the-shelf language models (GPT-3)

- I am planning a 7-day trip to Seattle. Can you make a detailed plan for me?
- Is there anything I can eat for breakfast that doesn't include eggs, yet includes protein and has roughly 700-1000 calories?
- Given a set of numbers find all possible subsets that sum to a given number.
- Give me a phrase that I can use to express I am very happy.

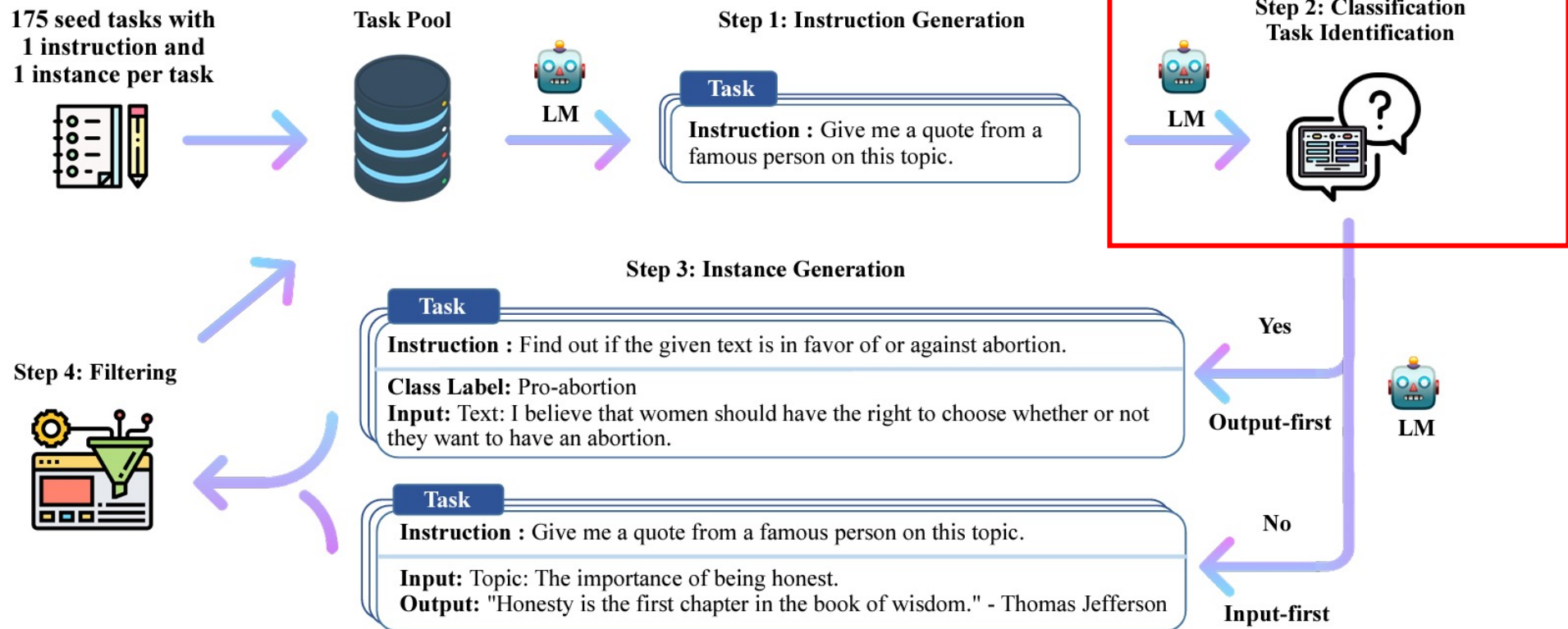
LM

Pre-trained, but **not aligned yet**

- Create a list of 10 African countries and their capital city?
- Looking for a job, but it's difficult for me to find one. Can you help me?
- Write a Python program that tells if a given string contains anagrams.

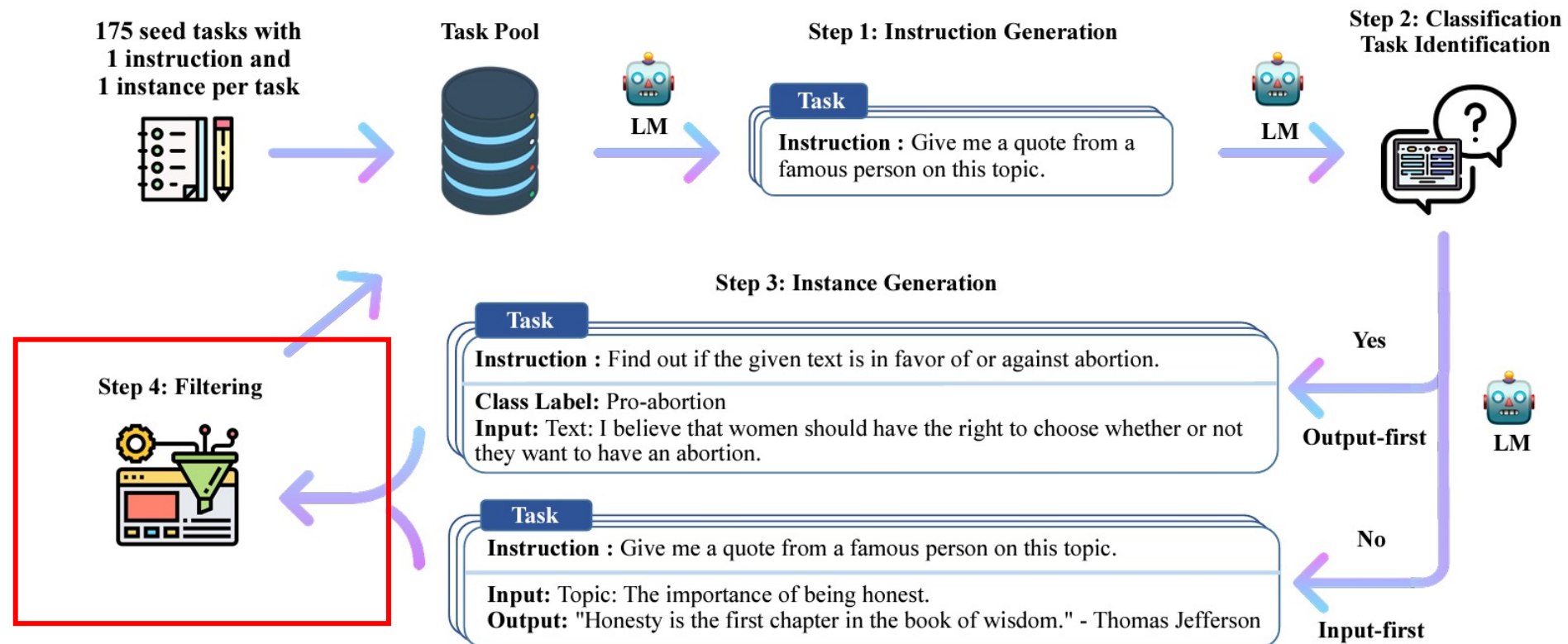
Self-Instruct Framework

- 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 ...



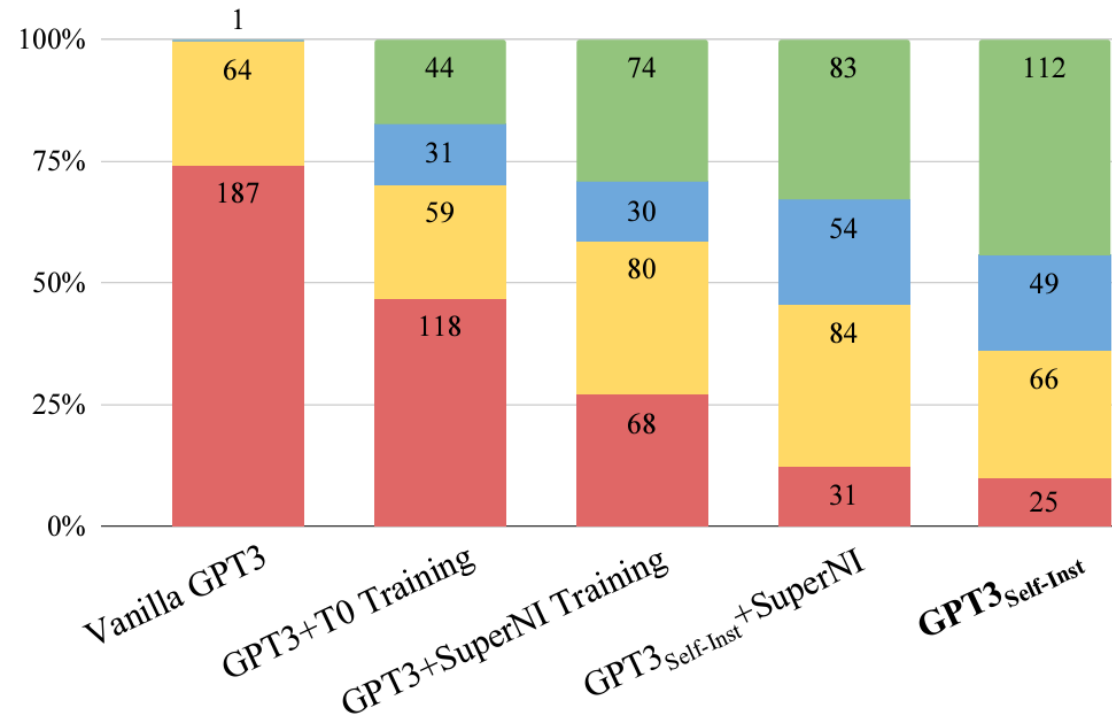
Experiment Results

- Use a GPT-3 (“davinci”) model to generate new instruction tasks, and fine-tune the GPT-3 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

Human Evaluation on User-Oriented Instructions

■ **A**: correct and satisfying response ■ **B**: acceptable response with minor imperfections
■ **C**: responds to the instruction but has significant errors ■ **D**: irrelevant or invalid response



- Self-training the model by bootstrapping instruction tasks from limited human-written seed tasks can improve model alignment

Content

- Instruction Tuning Overview
- Instruction Tuning on Public NLP Datasets
- Instruction Tuning on Crowdsourced Datasets
- Instruction Tuning on LM-Generated Datasets
- **Instruction Tuning on Mixture of Datasets**

LIMA: Less Is More for Alignment (Zhou et. al, 2023)

- Can we use a small number of data to instruct-tune a model to generalize to new tasks?
- Hypothesis: A model's knowledge and capabilities are learnt almost entirely during pre-training, while alignment teaches it the right format to be used when interacting with users

LIMA: Less Is More for Alignment (Zhou et. al, 2023)

- 1000 training examples: no more distillation data and with minor human annotations (200)
 - 750 top questions selected from community forums
 - manually write 250 examples of prompts and responses to emphasize the response style of an AI assistant
 - Finally train a 65B LLaMA model on 1000 demonstrations.

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

LIMA: Less Is More for Alignment

- Quality and diversity are the keys
- Quality Control:
 - Public data: select data with higher user ratings
 - In-house authored data: uniform tone and format
- Diversity Control:
 - Public data: Stratified sampling to increase domain diversity
 - In-house authored data: Increase task/scenario

Comparing LIMA with other LLMs

- Ask human crowd workers and GPT-4 which model response is better

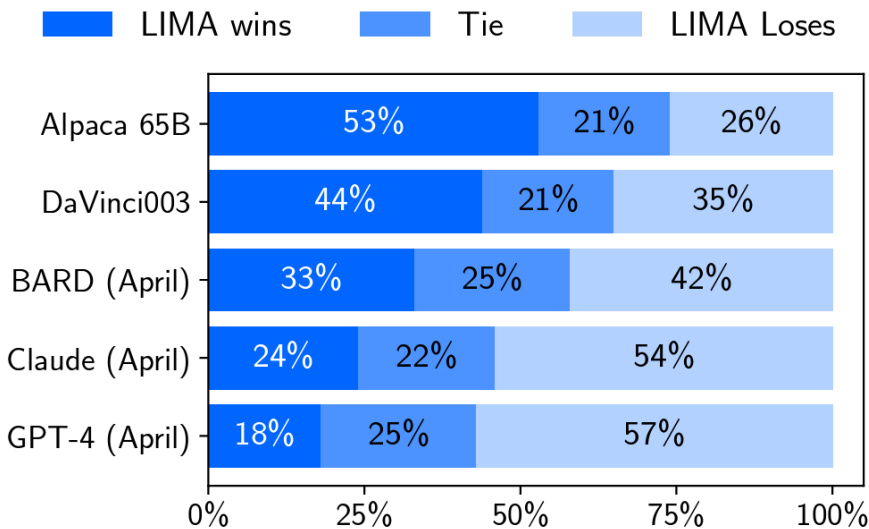


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

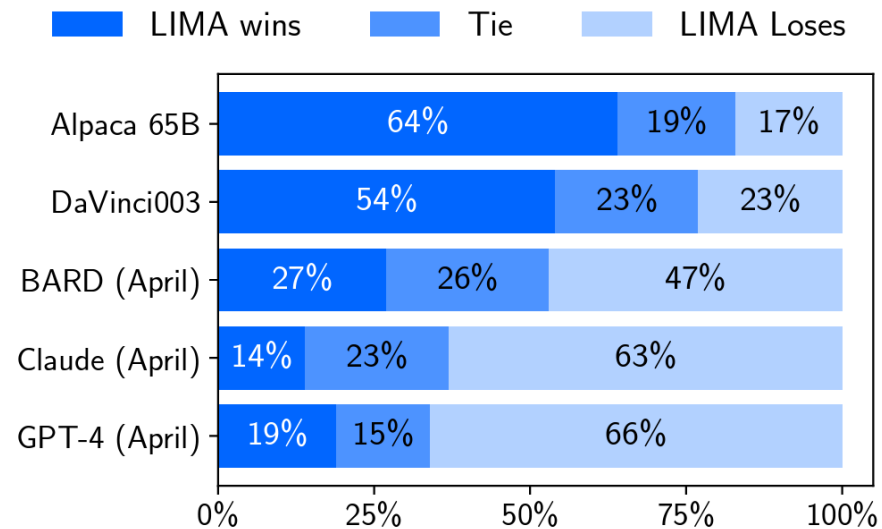


Figure 2: Preference evaluation using GPT-4 as the annotator, given the same instructions provided to humans.

Quality vs. Quantity vs. Diversity

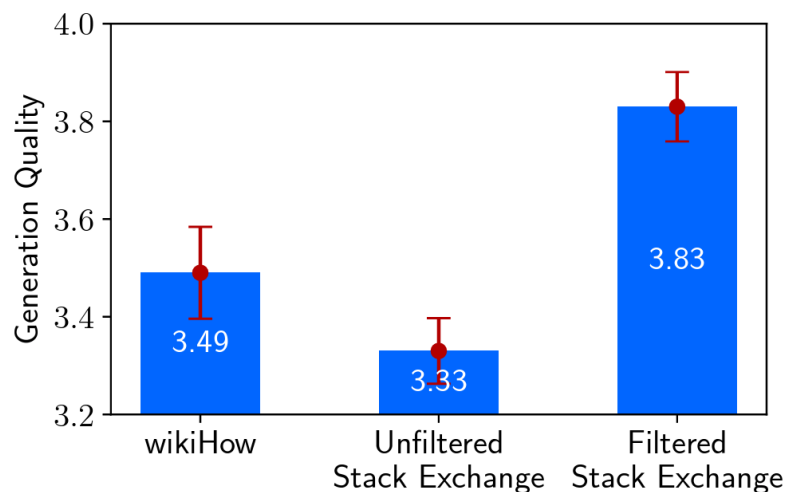
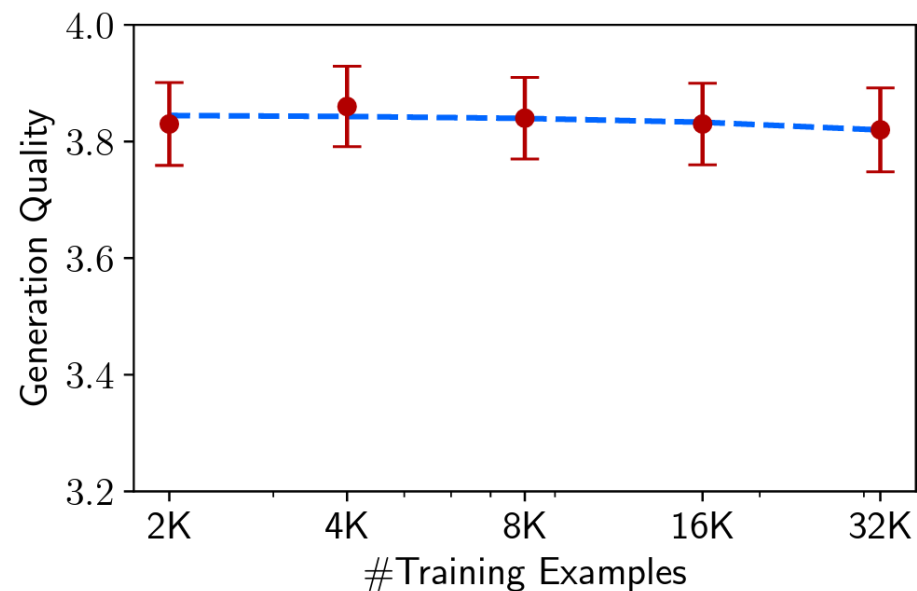


Figure 5: Performance of 7B models trained with 2,000 examples from different sources. **Filtered Stack Exchange** contains diverse prompts and high quality responses; **Unfiltered Stack Exchange** is diverse, but does not have any quality filters; **wikiHow** has high quality responses, but all of its prompts are “how to” questions.

- Scaling up training data does not necessarily improve the model response quality



Quality of Long-Form Highly Structured Response

- LIMA w. or w/o 6 format constraint examples
 - generating a product page with *highlights, about the product and how to use*
 - paper reviews with *summary, strengths, weakness 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

Tulu: How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources (Wang et. al, 2023)

- A comprehensive study on different instruction-tuning datasets
- Two mixtures of datasets
 - Human data mixture
 - Human + GPT data mixture

Comparison of Using Different Instruction Tuning 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









Different Base Models

- Base model quality is extremely important for downstream 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

Different Model Sizes

- Smaller models benefit more from instruction-tuning
- Instruction-tuning does not help to enhance strong capabilities already exist 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

Next Class: Language Model Reasoning

