# CS 546 – Advanced Topics in NLP

Dilek Hakkani-Tür

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

CONVAI

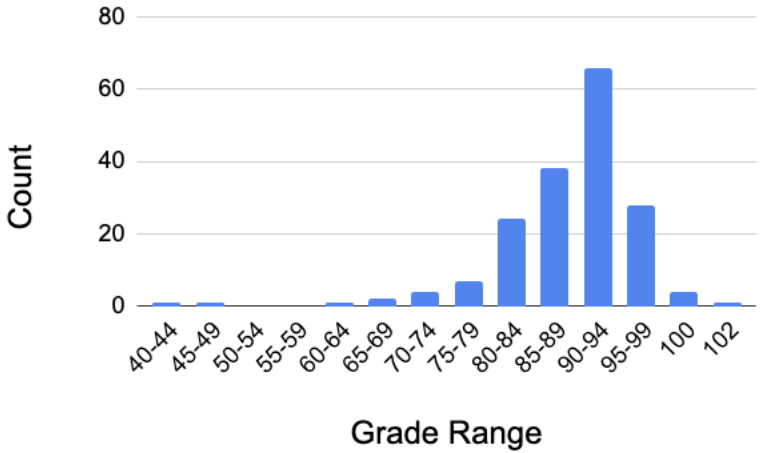
CDS

Siebel School of Computing and Data Science

# Midterm 1 Outcomes

Highest grade: 102/100 (2 points bonus)
Mean: 88.6
Median: 90

Midterm 1 is 15% of the overall grade.



# Topics for Today


Instruction Tuning

- Overview
- Instruction Tuning Datasets
- Evaluation
- Other Recent Instruction-Tuning Related Work

# Readings


- Zhang et al, Instruction Tuning for Large Language Models: A Survey, 2023 (updated till recently).

- https://github.com/RenzeLou/awesome-instruction-learning

# Instruction Tuning

- Mismatch between the training objective and the user's objective: Language modeling pre-training objective of predicting the next token does not align well with actual LLM use cases.
  - People are not only using LLMs for text completion.
  - They want the models to follow their instructions accurately and safely.

| Language Modeling Objective | ✖ | Helpful LMs |
| --- | --- | --- |
| Predicting the next token in a document | | Follow the user's instructions helpfully and safely |

Misaligned

# Instruction Tuning (cont.)

- Before instruction tuning

- Example output of GPT-3, from https://openai.com/index/instruction-following/

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.
```

# Instruction Tuning (cont.)

- Also called supervised fine-tuning or SFT

- Aims to steer models towards accurately (and safely) responding to real user requests.

- Further training LLMs using instruction and output pairs:
  - Instruction: human instruction for the models
  - Output: desired output that follows the instruction

# The Turking Test (Efrat and Levy, 2020)

- Can language models understand instructions?
- Introduced an instruction understanding task (IUT).
- A model is provided with an input $I_x$ that describes in natural language a desired output $o$.
- $I_x$ is comprised of a template $I$ called instruction, which is instantiated with a resource $x$ to form $I_x$.
- $x$ is highlighted in green in the example figure.

---

Write questions about the highlights of a story.

Steps
1. Read the highlights
2. Write questions about the highlights

Example

Highlights
• Sarah Palin from Alaska meets with McCain
• Fareed Zakaria says John McCain did not put country first with his choice
• Zakaria: This is "hell of a time" for Palin to start thinking about national, global issues

Questions
The questions can refer directly to the highlights, for example:
• Where is Palin from?
• What did Fareed say about John McCain's choice?
• Who is thinking about global issues?

Questions must always be related to the highlights but their answers don't have to be in the highlights. You can assume that the highlights summarize a document which can answer other questions for example:
• What was the meeting about?
• What was McCain's choice?
• What issues is Palin thinking about?

# The Turking Test (Efrat and Levy, 2020) (cont.)

- Can language models understand instructions ?
- Introduced an instruction understanding task (IUT).
- A model is provided with an input $I_x$ that describes in natural language a desired output $o$.
- $I_x$ is comprised of a template $I$ called instruction, which is instantiated with a resource $x$ to form $I_x$.
- $x$ is highlighted in green in the example figure.

Other Rules
• Do not re-use the same or very similar questions.
• Questions should be written to have short answers.
• Do not write "how" nor "why" type questions since their answers are not short. "How far/long/many/much" are okay.

Here are the highlights:
• Math geeks and others celebrate Pi Day every March 14
• Pi, or roughly 3.14, is the ratio of circumference to diameter of a circle
• The Pi Day holiday idea started at the Exploratorium museum in San Francisco
• Albert Einstein was also born on March 14

Write questions about them:

---

Here are questions about the highlights:
1. When is Pi Day celebrated?
2. What is the value of Pi up to the second decimal digit?
Another thing is that Pi is important in mathematics.

---

# InstructGPT (Ouyang et al, 2022)

- Making language models bigger does not inherently make them better at following a user's intention.

- Goal: align language models with user's intentions

- Intentions:
  - Explicit: following user instructions
  - Implicit: staying truthful, and not being biased, toxic, or otherwise harmful
- 3H-objective:
  - Helpful: they should help user solve their task
  - Honest: they shouldn't fabricate information or mislead the user
  - Harmless: they should not cause physical, psychological, or social harm to people or the environment

- Method: fine-tuning pre-trained LMs with human feedback

# InstructGPT Approach

CONVAI

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

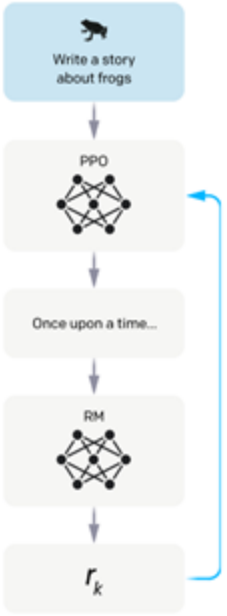
**Optimize a policy against the reward model using reinforcement learning.**

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

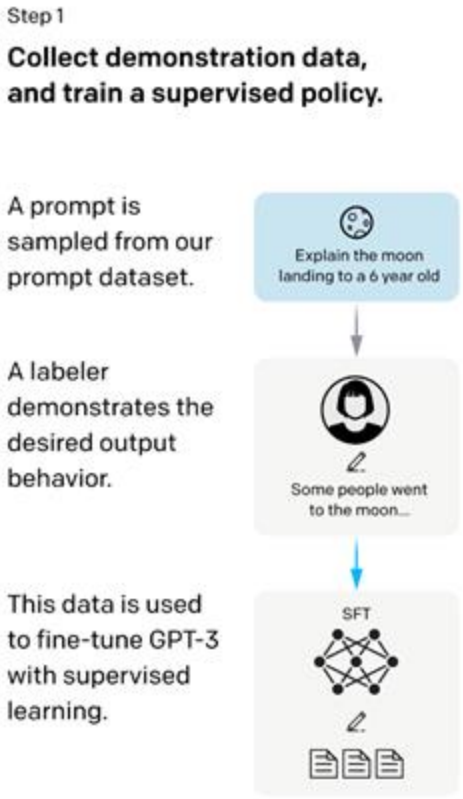
The reward is used to update the policy using PPO.



# InstructGPT - Step 1, supervised fine-tuning (SFT)

- Fine-tune GPT-3 using supervised learning
- Data collection:
  - Collecting Prompts:
    - Contractor labelers (40 trained experts)
    - Prompts submitted to the InstructGPT API by customers
  - Asking labelers to write high-quality responses

| SFT Data | | |
|---|---|---|
| split | source | size |
| train | labeler | 11,295 |
| train | customer | 1,430 |
| valid | labeler | 1,550 |
| valid | customer | 103 |



# Prompt Examples

- Generation: Write a short story where a brown bear to the beach, makes friends with a seal, and then return home.
- Brainstorming: List five ideas for how to regain enthusiasm for my career
- Classification: {java code} What language is the code above written in?
- Extract: Extract all place names from the article below: {news article}
- Open QA: Who built the statue of liberty?

| Use-case | (%) |
|---|---|
| Generation | 45.6% |
| Open QA | 12.4% |
| Brainstorming | 11.2% |
| Chat | 8.4% |
| Rewrite | 6.6% |
| Summarization | 4.2% |
| Classification | 3.5% |
| Other | 3.5% |
| Closed QA | 2.6% |
| Extract | 1.9% |

# InstructGPT - Step 2, Training the Reward Model

- RM (GPT-6B with an additional layer):
- Input: a prompt $x$ and a response $y$
- Output: a scalar reward r($x$,$y$)
- Loss function:
  - $K$: the total number of responses for the prompt
  - $y_w$, $y_l$: two responses of the prompt $x$, where $y_w$ is the preferred one

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta(x,y_w) - r_\theta(x,y_l)\right)\right)\right]$$

| RM Data | | |
|---|---|---|
| split | source | size |
| train | labeler | 6,623 |
| train | customer | 26,584 |
| valid | labeler | 3,488 |
| valid | customer | 14,399 |

K ranges from 4 to 9

# InstructGPT – Step 3, Refining the Model with RL

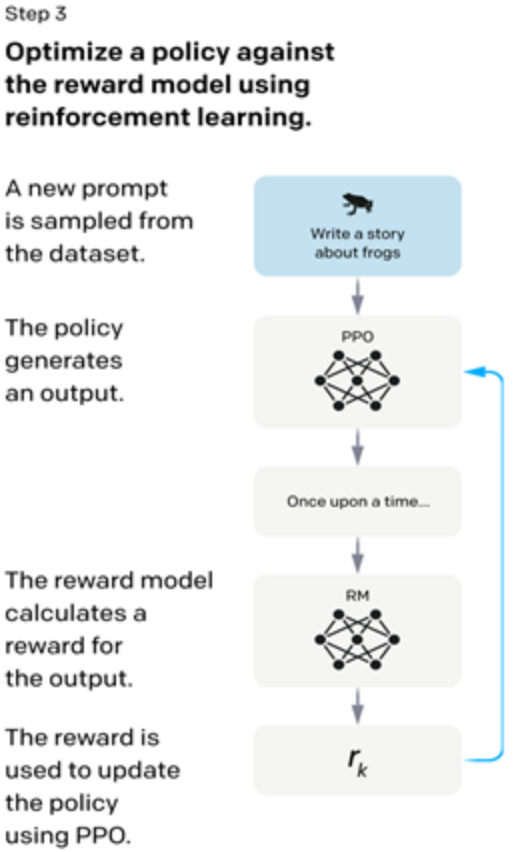
- Fine-tune the SFT model using reinforcement learning

- Objective function:
  - Reward for y obtained by the RM
  - KL penalty to make sure the generated output is not too far away from the output of SFT

More on alignment and reinforcement learning by Ishika in a few weeks!

PPO Data

| split | source | size |
|---|---|---|
| train | customer | 31,144 |
| valid | customer | 16,185 |



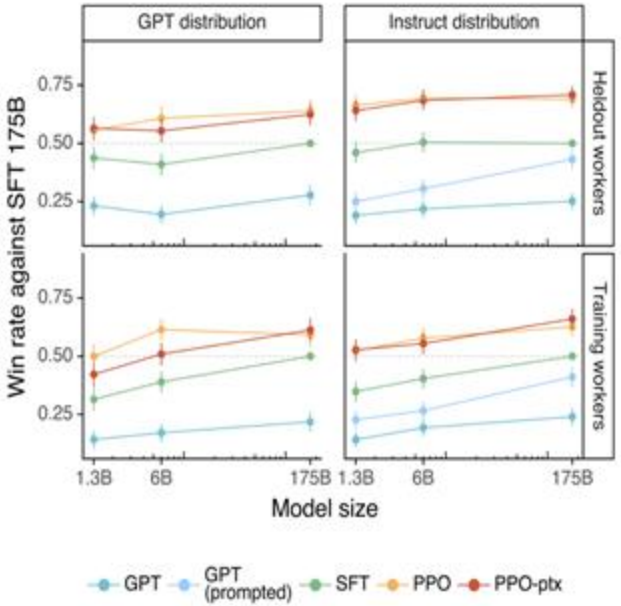
# InstructGPT – Evaluation

- Evaluations on API distribution:
  - Dataset:
    - prompts submitted to InstructGPT API that haven't been included in training
    - Prompts submitted to GPT-3 API
  - Metrics:
    - human preferred rate (winrate) compared with 175B SFT
    - 1-7 Likert score of the overall quality of the outputs, as well as their binary metadata

- Evaluations on public NLP datasets:
  - Measuring LM safety:
    - TruthfulQA, RealToxicityPrompts, …
  - Measuring zero-shot performance on traditional NLP tasks, such as question answering, reading comprehension, and summarization
    - Hellaswag, SquadV2, …

# InstructGPT – Evaluation (cont.)

- Labelers significantly prefer InstructGPT outputs over outputs from GPT-3.



# InstructGPT - Qualitative Example

**Prompt:**
What is the purpose of the list C in the code below?

```
def binomial_coefficient(n, r):
    C = [0 for i in range(r + 1)];
    C[0] = 1;
    for i in range(1, n + 1):
        j = min(i, r);
        while j > 0:
            C[j] += C[j - 1];
            j -= 1;
    return C[r]
```

**GPT-3 175B completion:**
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]

**InstructGPT 175B completion:**
The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

# A challenge for Instruction Tuning

- It is expensive and labor-intensive to collect such data
  - Often times you need expert responses or detailed research to form the answer, e.g.,

    “I have taken 80 credits from CS and 12 credits from Linguistics. Can I do a minor in Linguistics at UIUC?”

  - Tasks that require creative generation do not have one right answer, e.g.,

    ”Write a poem to celebrate the birthday of my sister. She likes elephants.”

# Topics for Today


Instruction Tuning

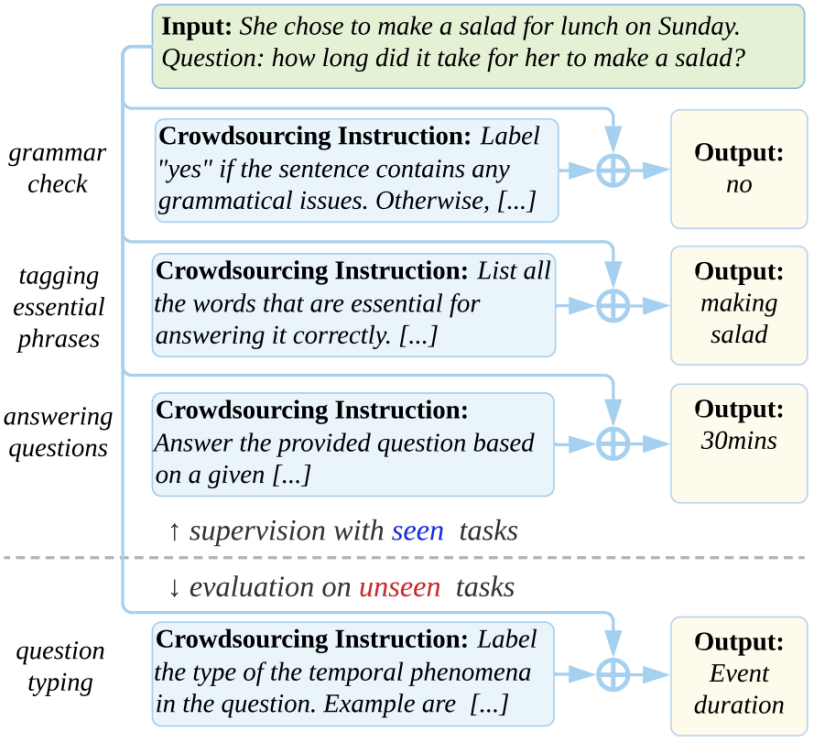
- Overview
- Instruction Tuning Datasets
- Evaluation
- Other Recent Instruction-Tuning Related Work

# Forming Instruction Datasets

- Manually created datasets
- Synthesizing data using strong LLMs
- Synthesizing data using the same model

# NATURAL INSTRUCTIONS Dataset

- (Mishra et al, ACL, 2022)
- Cross-task generalization benchmark and dataset.
- 61 distinct NLP tasks, 193 K instances
- Instruction tuning models with existing datasets can help with generalization towards unseen tasks.



# NATURAL INSTRUCTIONS Dataset – Example

CONVAI

### 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?

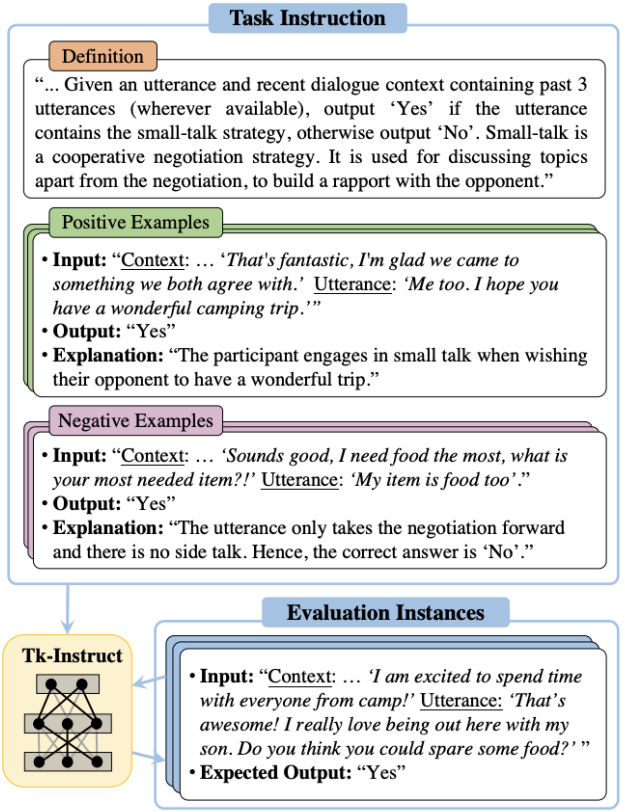
⋮

**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?

# Super-Natural Instructions

- (Wang et al, EMNLP 2022)
- Extensive cross-task generalization.
  - Multi-lingual instruction collection (55 languages)
  - Work of a much larger consortium, following Natural Instructions.
  - 1,616 diverse NLP tasks, 5M task instances, 76 distinct task types

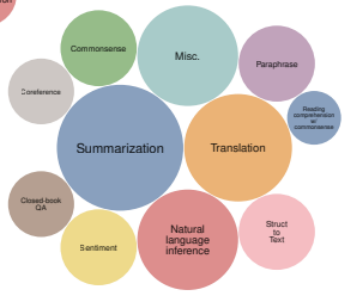


# Super-Natural Instructions – Data Analysis

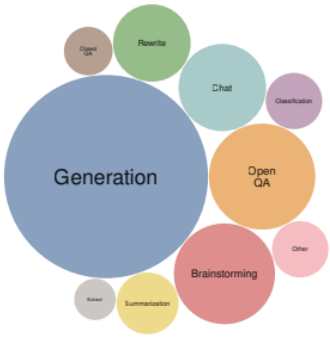

(a) SUP-NATINST (this work)


(b) NATINST


(c) PROMPTSOURCE (T0 subset)




(d) FLAN




(e) INSTRUCTGPT

# Super-Natural Instructions – Evaluation

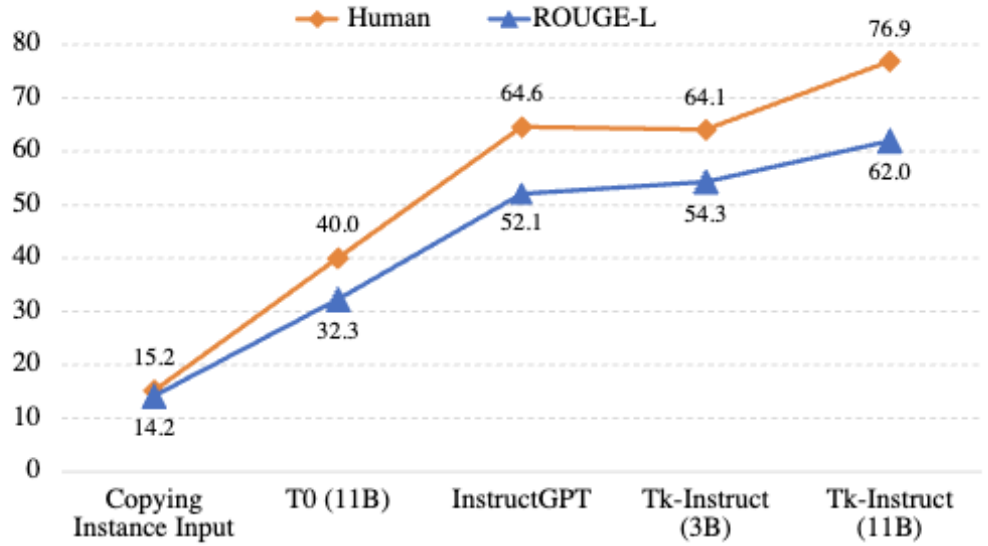


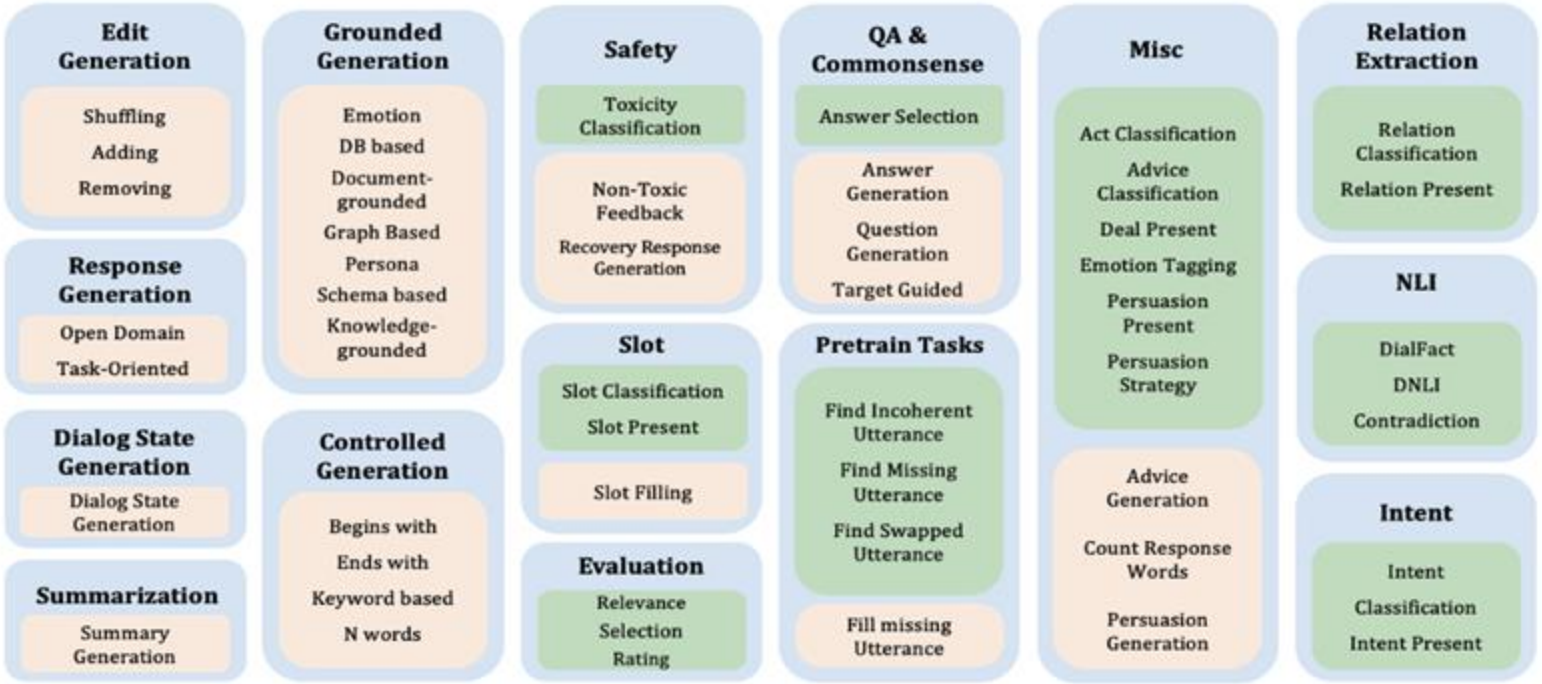

Figure 3: Human evaluation vs. ROUGE-L for several methods (§6.2). The trends of these two metrics are highly correlated with a Pearson coefficient of 0.998.

- Instruction-tuning enables stronger generalization to unseen tasks
- There is also sizable gap for improvement in comparison to supervised training.

# InstructDial

- (Gupta et al., EMNLP 2022)
- Specialized to dialogue related tasks that require conversational data.
- Instruction tuning for 48 dialogue tasks, 59 datasets
- Introduces novel **meta-tasks** (e.g. select an instruction that matches with an input-output pair) to encourage models to adhere to the instructions.

# InstructDial – Task Taxonomy




**Green** represents classification and **orange** represents generation tasks.

# InstructDial – Examples

CONVAI

## Relation classification

**Instruction**: You will be given some conversation text and you need to find the relation in the conversation between specified people or speakers.

**Input**: **[CONTEXT]** Speaker 1: You know Phoebe, when I was little... **[ENDOFTURN]** Speaker 2: Oh, I love family ... **[ENDOFDIALOGUE]** Possible relations are: **[OPTIONS]** 0: students, 1: visited place, 3: schools attended, 4: siblings, .... **[QUESTION]** Choose the most possible relation between Speaker 1 and Ursula

**Output**

4

## Emotion grounded generation

**Instruction**: In this task, write a response to the conversation so that the response contains the emotion provided

**Input**: **[EMOTION]** joy **[CONTEXT]** Oh God, what happened? **[ENDOFTURN]** Oh. God, crazy Chandler. He spun me off the bed! **[ENDOFDIALOGUE]** **[QUESTION]** Given the context and emotion, the response is

**Output**

Wow! Spinning that sounds like fun.

## Instruction selection

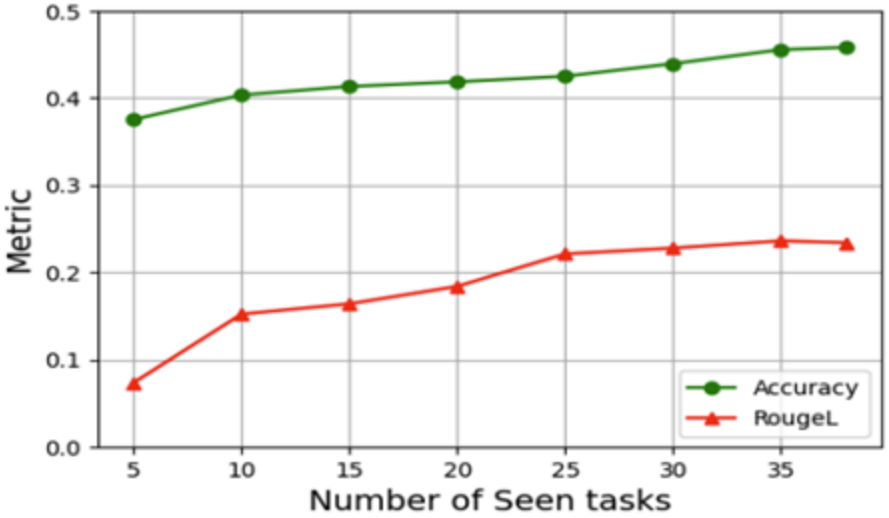
**Instruction**: In this task given a dialogue input and the output corresponding to a task for the dialogue, choose the best instruction for the task among the provided options

**Input**: **[CONTEXT]** Are you buying a house? **[RESPONSE]** I like to see what is available . **[ENDOFDIALOGUE]** The output is: **[OUTPUT]** 8. The list of options for instructions are: **[OPTIONS]** A: find intent, B: find length of response, C: Find value of slot name **[QUESTION]** The correct option among instructions is:

**Output**

B

# InstructDial – Generalization to Unseen Tasks




- Training on more seen tasks improves generalization on **unseen tasks**.

# Less is More for Alignment (LIMA)

- (Zhou et al., Neurips, 2023)
- Measure the relative importance of pre-training and instruction fine-tuning.
- LIMA, a 65B parameter LLaMa language model fine-tuned with the standard supervised loss on only 1,000 carefully curated prompts and responses, without any reinforcement learning or human preference modeling.
- Even a small amount of carefully selected high-quality instruction data can significantly improve model performance through instruction tuning.

# Manually Created Datasets

**Pros:**

- Annotation quality can be high.

**Cons:**

- Diversity of the examples can be limited.
- Tasks may be more focused to NLP than real user instructions.

# Forming Instruction Datasets

- Manually created datasets
- Synthesizing data using strong LLMs
- Synthesizing data using the same model

# Unnatural Instructions

- (Honovich et al., ACL, 2023)
- Collecting data with instructions and accurate responses is challenging.
- Previous work:
  - **Reformulated existing datasets. E.g.,** used prompt engineering to reformulate the datasets in instruction format: instruction—input—output.
    - The tasks and contents are limited by the diversity of the existing datasets.
  - **Human annotation** Collected user-generated prompts and manually annotated their expected outputs.
    - Requires live LLMs with existing users.
    - High cost of human labor.

# Unnatural Instructions (cont.)

- Goal: Instruction datasets with diverse tasks and contents, obtained with minimum human annotation.
- $M$: OpenAI's text-davinci-002 (part of GPT3.5 series, trained with SFT on human data).
- Approach Overview:

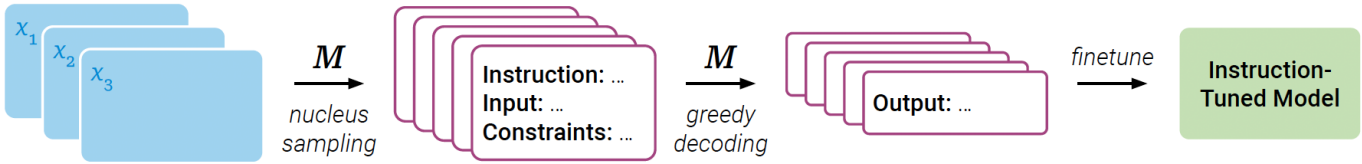
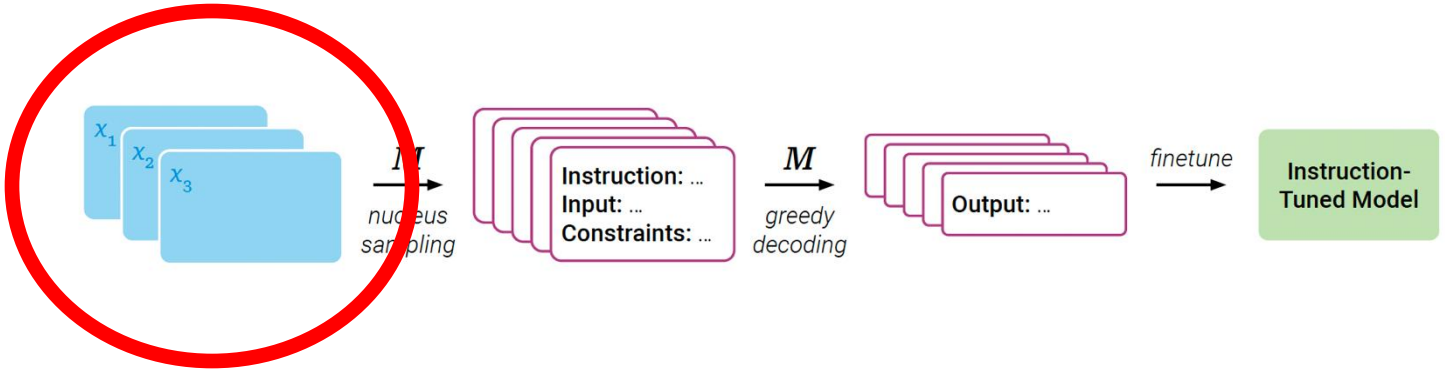



Figure 2: The core Unnatural Instructions generation pipeline. We use a seed of three in-context demonstrations $x_1, x_2, x_3$ to create a large dataset of NLP tasks with instructions, inputs and outputs. As a first step, we sample instructions, inputs, and constraints from a language model $M$. In the next step, we use $M$ to deterministically generate the corresponding outputs. Finally, the data can be used for instruction tuning.

# Unnatural Instructions – Seed Instruction Selection




- 15 manually annotated examples in the format of instruction—input—constraints.
- 3 in-context examples as input, 5 different seeds.

**Example 1**

**Instruction:** You are given a science question (easy-level) and four answer options (associated with "A", "B", "C", "D"). Your task is to find the correct answer based on scientific facts, knowledge, and reasoning. Do not generate anything else apart from one of the following characters: 'A', 'B, 'C', 'D'. There is only one correct answer for each question.

**Input:** Which part of a bicycle BEST moves in a circle? (A) Seat (B) Frame (C) Foot pedal (D) Kickstand

**Constraints:** The output should be one of the following characters: 'A', 'B, 'C', 'D'.

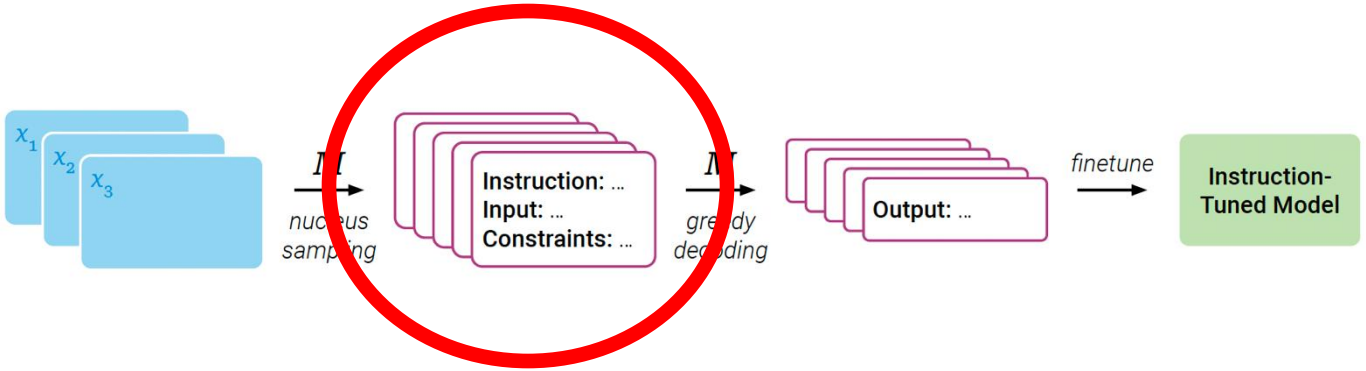
**Example 2**

**Instruction:** You are given a negative review and your task is to convert it to a positive review by one or more making minimal changes. Avoid changing the context of the review.

**Input:** we stood there in shock, because we never expected this.

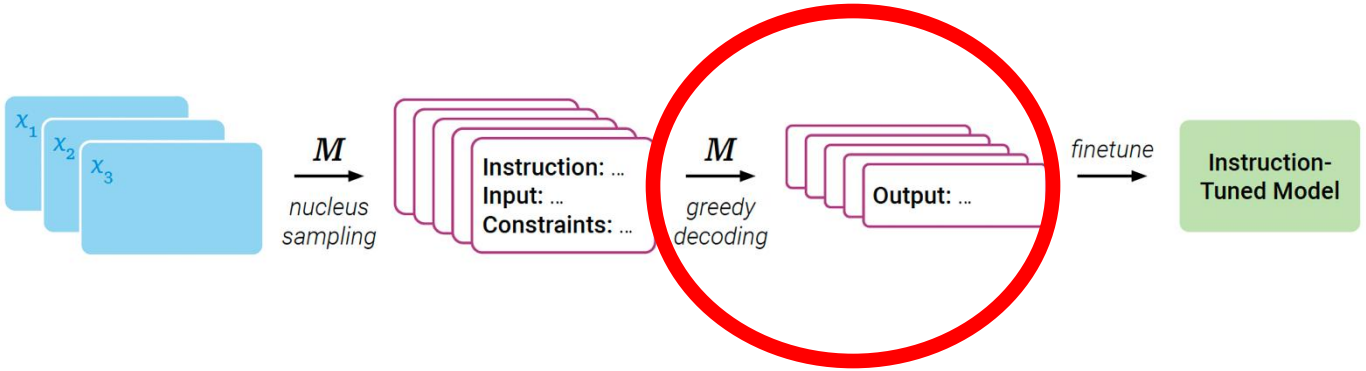
**Constraints:** None.

# Unnatural Instructions – Core Dataset Generation




- Use LLM generate a large number of training samples, based on the in-context examples.
- Nucleus sampling for diversity of samples.

# Unnatural Instructions – Core Dataset Generation

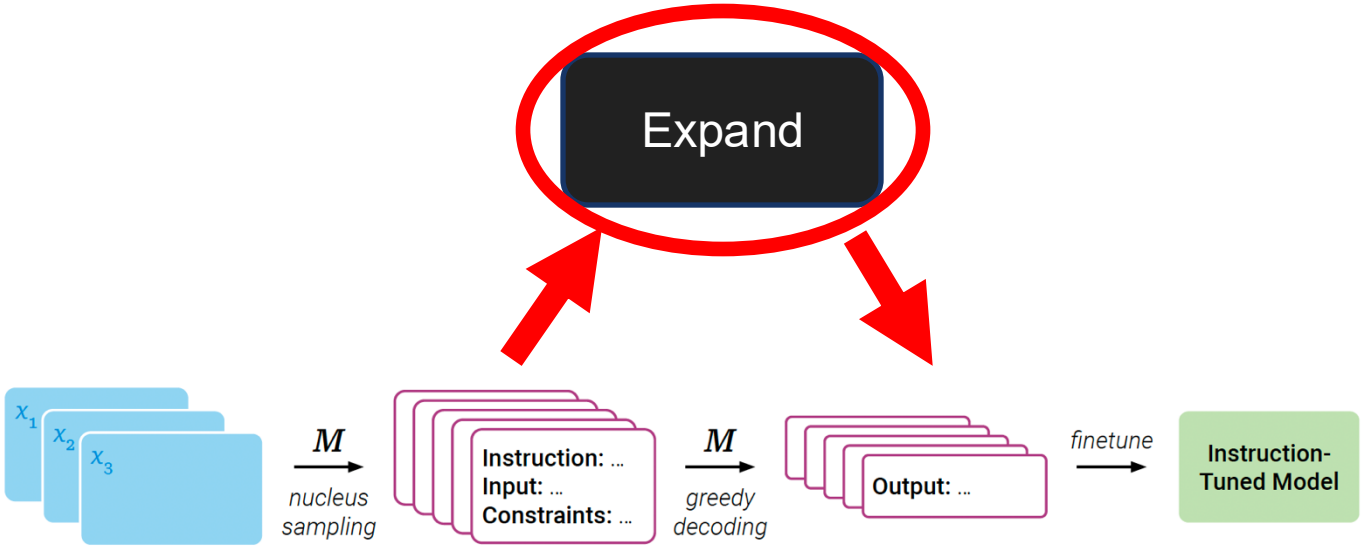



- LLM is used for generating instruction outputs as well.
- Greedy decoding for accuracy when generating the outputs.

# Unnatural Instructions – Core Dataset Filtering


- Not all generated examples are good.
- Manual analysis of 200 generated examples showed only 113 (56.5%) are accurate.
  - Of the 87 incorrect examples, 9 (4.5%) had incomprehensible instructions, 35 (17.5%) had an input that did not match the task description, and 43 (21.5%) had incorrect outputs.
- Filtering Strategy
  - Violation of the format of instruction—input—constraints.
  - Identical items as the seed examples.
  - Duplicates.

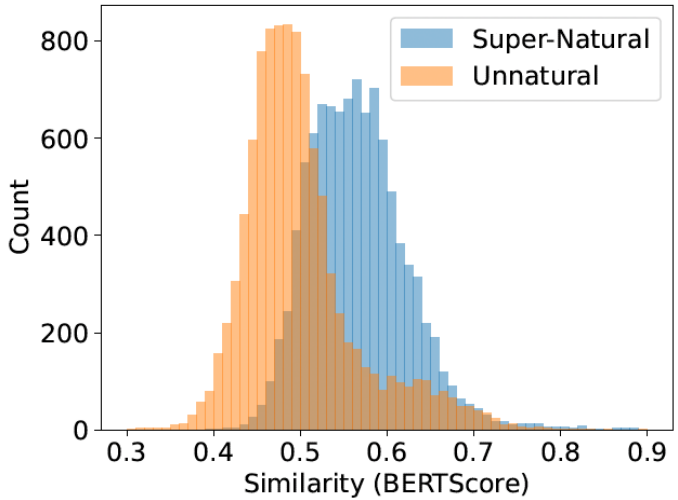
# Unnatural Instructions – Template Expansion




- Reformulate the instructions through adding {INPUT} placeholder and rephrasing via LLM.
- Few-shot learning with in-context examples, similar to the core dataset generation.
- Core dataset: 58K examples, Reformulation: 240K examples

# Unnatural Instructions – Diversity Analysis

- Comparison with Super-Natural Instructions
- Similarity of 1000 randomly selected pairs in each dataset

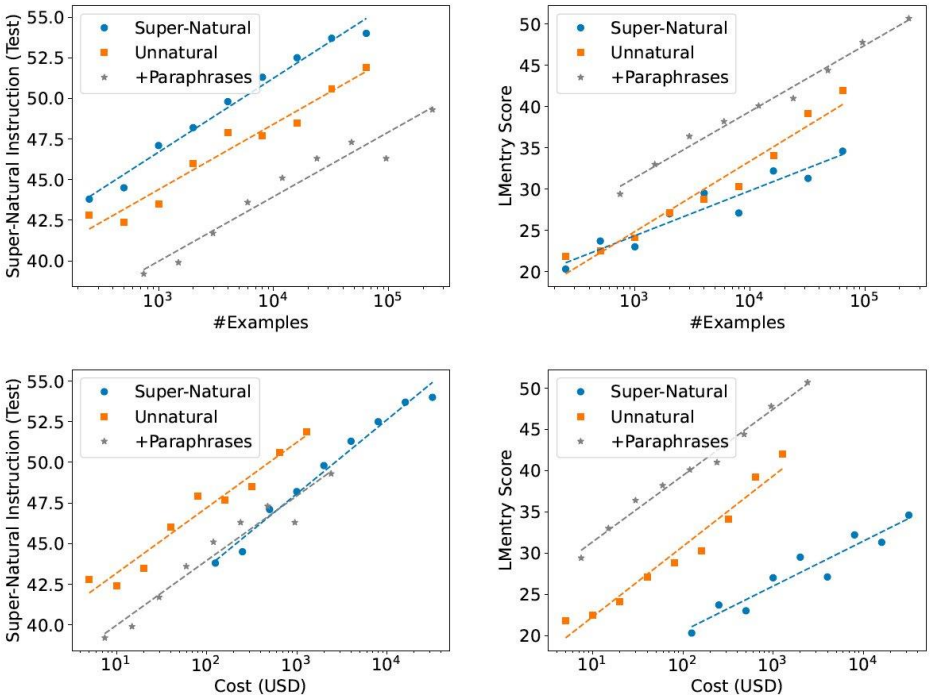


# Unnatural Instructions – Experiments

- Baselines:
  - T5-LM on Super-Natural Instructions: Direct comparison.
  - T0++, Tk Instruct, FLAN–T5…, which are all instruction tuned T5 models.

| Model | #Examples | Super-Natural Instructions | T0: Zero-Shot | BIG-bench: Hard (Orig/QA) | LMentry |
|---|---|---|---|---|---|
| ***Prior Work*** | | | | | |
| T5-LM | 0 | 24.3 | 40.2 | 0.0 / 0.7 | 20.6 |
| T0++ | 12,492,800 | 40.3 | NHO | 20.2 / 13.9 | 38.3 |
| Tk-Instruct | 75,417 | 45.6 | 41.4 | 5.8 / 11.8 | 35.7 |
| FLAN-T5 | 14,336,000 | NHO | NHO | 39.3 / 40.0 | 52.2 |
| ***Direct Comparison Baseline*** | | | | | |
| T5-LM on Super-Natural Instructions | 64,000 | **54.0** | 44.0 | 10.2 / **29.7** | 34.6 |
| ***Our Approach*** | | | | | |
| T5-LM on Unnatural Instructions | 64,000 | 51.9 | 45.7 | 16.0 / 29.5 | 42.0 |
| + Instruction Paraphrases | 240,670 | 49.3 | **49.0** | **28.1** / 29.4 | **50.7** |

# Unnatural Instructions – Scaling Analysis




- Based on OpenAI's pricing as of December 2022, the cost for generating an example is estimated at $0.02 for the core dataset, and $0.01 for the expanded dataset.
- Kiela et al. (2021) estimate human annotation cost at $0.50–$1.00 per example.

# Forming Instruction Datasets

- Manually created datasets
- Synthesizing data using strong LLMs
- Synthesizing data using the same model

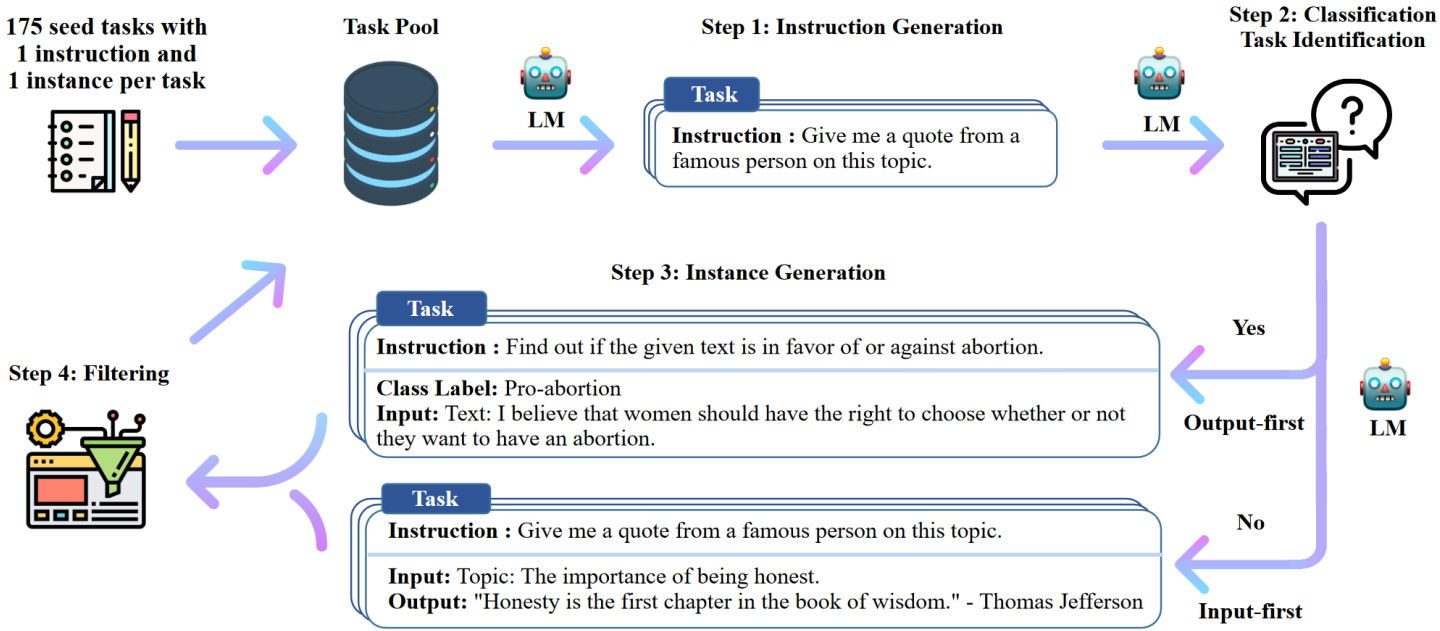
# Self-Instruct

- (Wang et al., ACL 2023)

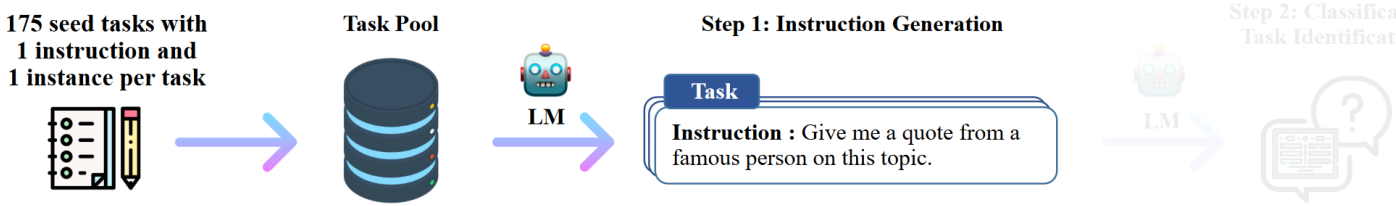
- Fine-tuning on human-generated instruction data is challenging:
  - Collecting instruction data is costly
  - Limited diversity because most them tend to be popular NLP tasks

- Iterative solution
  - Use vanilla language model (e.g., GPT3) to generate instruction data
  - Fine-tune the model on generated instruction data, bootstrapping off its own generations.

# Self-Instruct – Approach



# Self-Instruct – Approach, Step 1




- Recap: a task contains 1 instruction and many instances (input-output pairs)
- Step 1: generate instructions
- Method
  - In-context learning
  - Sample tasks from task pool of 175 tasks as few-shot examples

```
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:
```

Prompt used to generate new instructions

# Self-Instruct – Approach, Steps 2&3

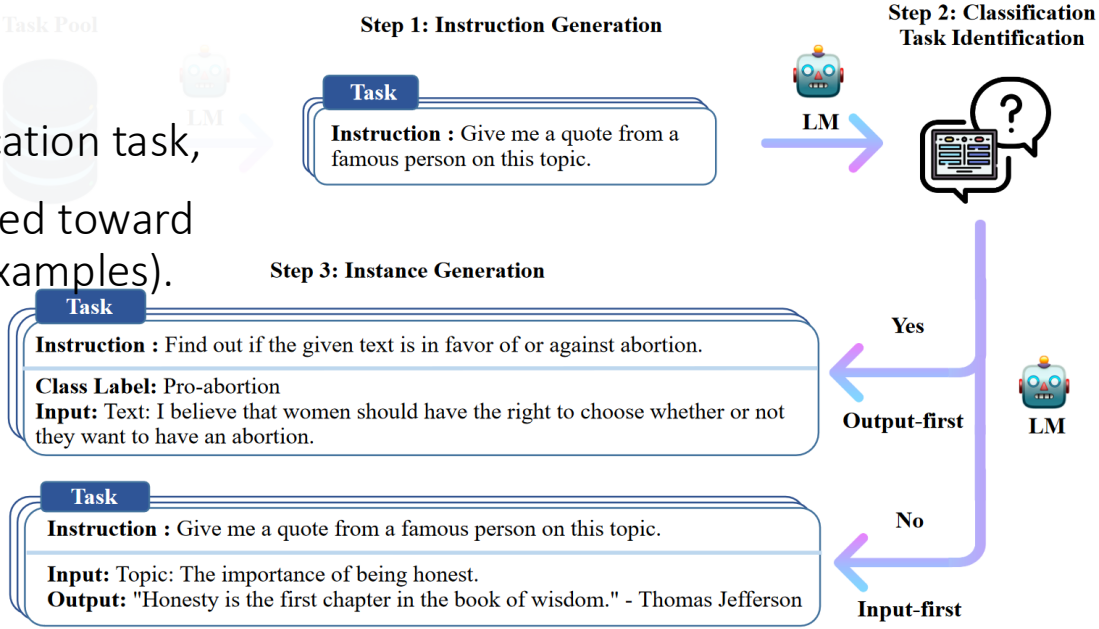
- The next step is to generate input-output pairs for a task
- Again, in-context learning
- However, if the task is a classification task,

the generated input can be biased toward one label (i.e., only grammatical examples).

-> output-first can be a solution.

- Solution
  - Identify classification tasks
  - For a classification task,

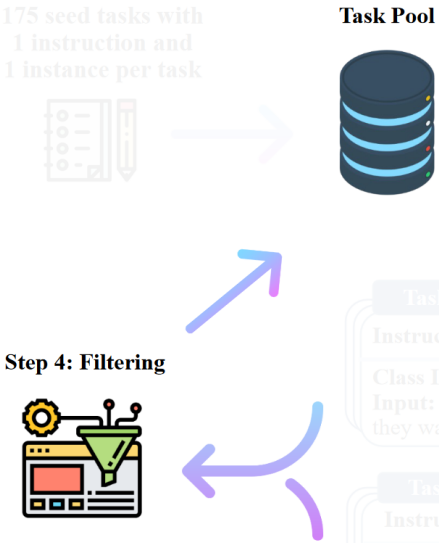
generate output (class label) first



# Prompt: Classification Task Identification

```
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:  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}
```

# Self-Instruct – Approach, Step 4




Filter instructions and instances to improve quality/diversity of the data:

Instruction:

- Too long/short
- Similar to existing instructions (ROUGE-L > 0.7) (ROUGE-L is a longest common subsequence-based text similarity metric)
- Keywords: image, picture, graph

Instance:

- Duplicates
- Same input but different output
- Output is a repetition of the input

# Self-Instruct – Examples of Generated Instructions

**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 ...

LM: GPT3

# Self-Instruct – Data Stats

- 52k instructions, 82k instances (vs. 1616 instructions in SUPERNI)

| 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 |

Table 1: Statistics of the generated data by applying SELF-INSTRUCT to GPT3.

# Self-Instruct – Data Diversity

- Uses Berkeley Neural Parser to parse instruction and extracts verbs closest to the root and its direct object.
- 20 most common verbs in the portion of the data where such extraction was possible.

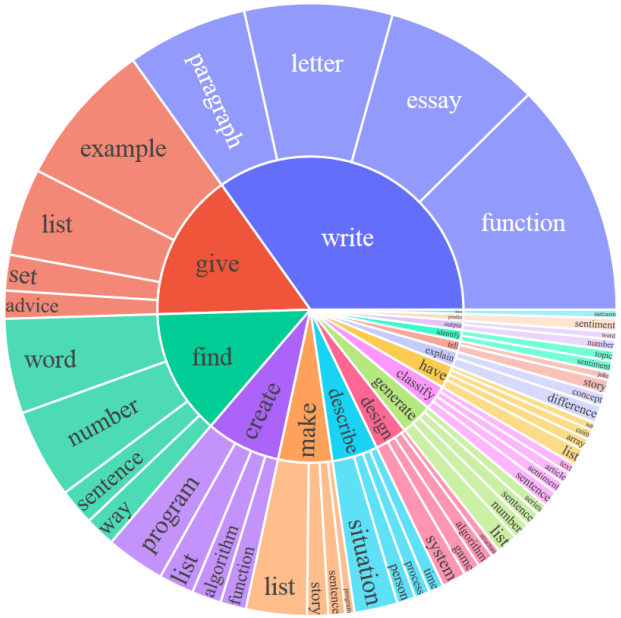



Figure 3: The top 20 most common root verbs (inner circle) and their top 4 direct noun objects (outer circle) in the generated instructions. Despite their diversity, the instructions shown here only account for 14% of all the generated instructions because many instructions (e.g., “Classify whether the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

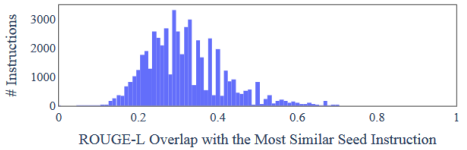



Figure 4: Distribution of the ROUGE-L scores between generated instructions and their most similar seed instructions.

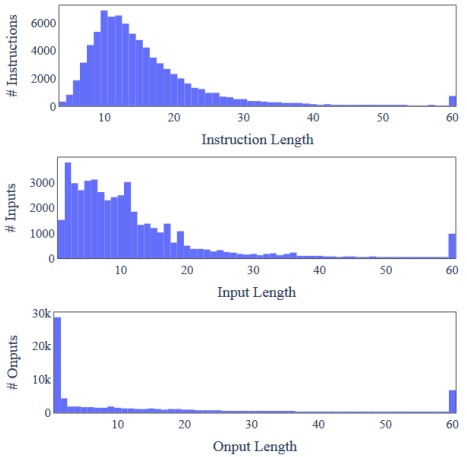



Figure 5: Length distribution of the generated instructions, non-empty inputs, and outputs.

# Self-Instruct – Data Quality

- Manually examined 200 instructions with 1 random instance per instruction.
- Error rate relatively high (46%)
- However, most of them are in correct format or partially right, which can still guide the model to follow instructions
- Paper doesn't have a detailed task breakdown for this analysis.

| Quality Review Question | Yes % |
|---|---|
| Does the instruction describe a valid task? | 92% |
| Is the input appropriate for the instruction? | 79% |
| Is the output a correct and acceptable response to the instruction and input? | 58% |
| All fields are valid | 54% |

Table 2: Data quality review for the instruction, input, and output of the generated data. See Table 10 and Table 11 for representative valid and invalid examples.

# Self-Instruct -- Performance

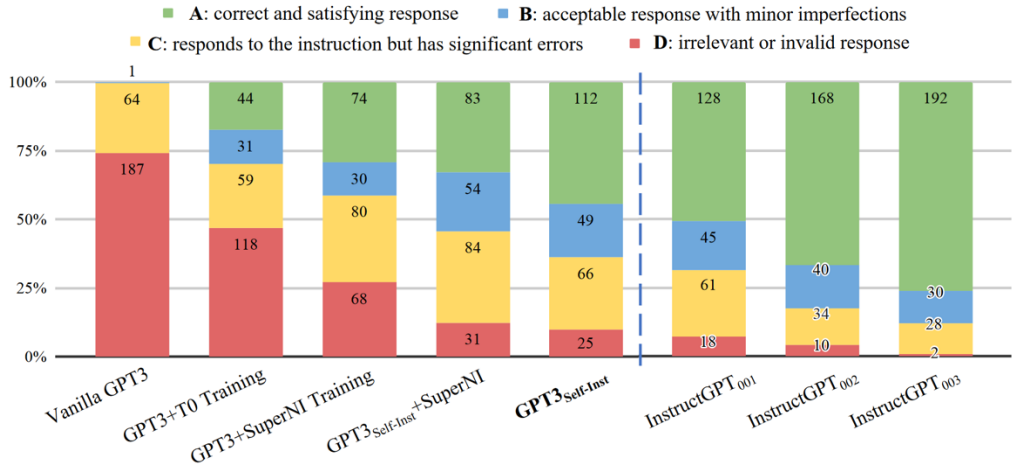
- Fine-tuning details: GPT-3 via OpenAI's fine-tuning API, 2 epochs
- Test Set:
  - SUPERNI test split. 119 tasks, each with 100 instances in each task, zero-shot generalization
- Observations
  - ① SELF-INST boosts GPT3 performance by a large margin
  - ② Nearly matches InstructGPT-001
  - ③ Brings additional gains when combined with SUPERNI training set
    - SUPERNI training and test sets have similar instruction style and formatting

| | 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 |
| ② | $\text{InstructGPT}_{001}$ | 175B | **40.8** |
| | **Instruction-tuned w/ SUPERNI** | | |
| | T$k$-INSTRUCT | 11B | 46.0 |
| ③ | GPT3 + SUPERNI Training | 175B | 49.5 |
| ③ | $\text{GPT3}_{\text{SELF-INST}}$ + SUPERNI Training (Ours) | 175B | **51.6** |

# Self-Instruct – Performance (cont.)

- Fine-tuning details: GPT-3 via OpenAI's fine-tuning API, 2 epochs
- Test Set:
  - A small dataset for human evaluation. 252 tasks, each with 1 instance.
  - Generalization to user-oriented instructions on novel tasks (e.g., email writing, social media, productivity tools, entertainment, programming)

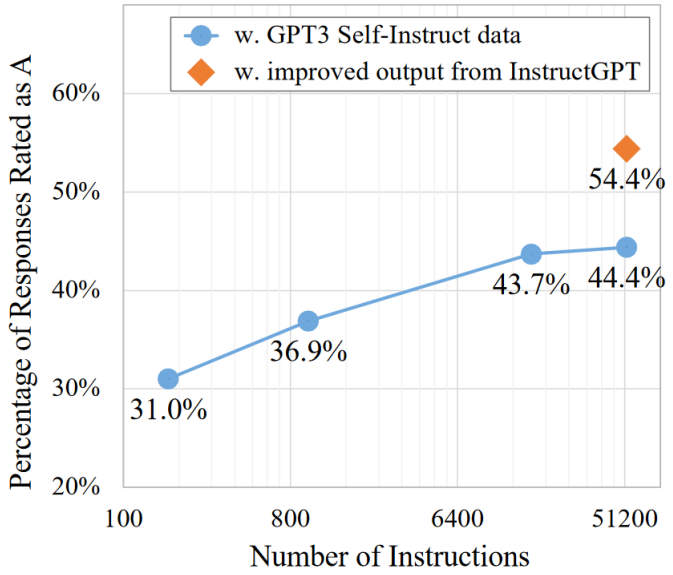
- Human evaluation
  - A: valid and satisfying
  - B: acceptable but has minor errors
  - C: responds to the instruction, but has significant errors
  - D: irrelevant or completely invalid



# Self-Instruct – Performance (cont.)

- Data size
  - Performance improves as the data size grows
  - The improvement almost plateaus after 16K
  - When evaluating on SUPER-NI, it plateaus at around hundreds of instructions
  - Possible reason: Generated data is distinct from typical NLP tasks in SUPER-NI
- Data quality
  - Regenerate the output field with InstructGPT003
  - 10% gain
  - Wish they did this exp. for all data points!

Human evaluation performance of $GPT3_{SELF\text{-}INST}$ models tuned with different sizes of instructions:



# Follow-Up Work


- Stanford Alpaca (March 2023)
  - Instruction-following LLaMA model fine-tuned using SELF-INSTRUCT
  - Minor modifications to the original SELF-INSTRUCT
    - Used text-davinci-003 (an instruction-tuned model)
    - Discarded the difference between classification and non-classification instructions
    - Generated single instance for each instruction
    - New instruction generation prompt
- Not use vanilla LM, but instruction-tuned LM to generate instruction data
- OpenHermes-2.5, a popular open-source instruction and chat dataset, containing mostly GPT-generated data
- OpenAI suspends ByteDance's account after it used GPT to train its own model

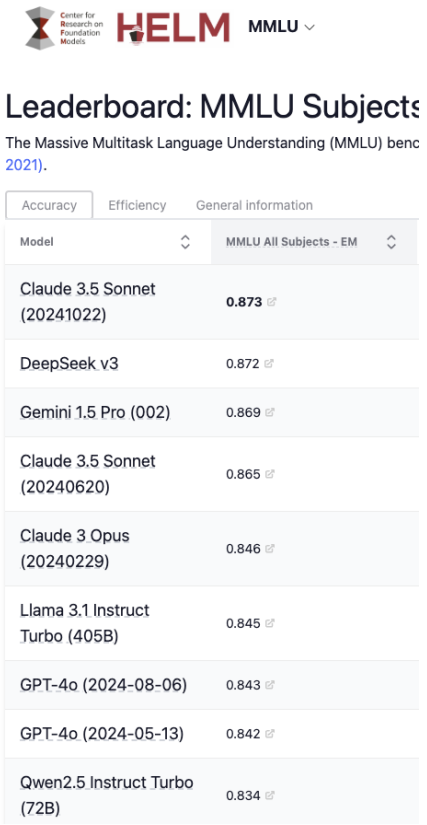
# Topics for Today


Instruction Tuning

- Overview
- Instruction Tuning Datasets
- Evaluation
- Other Recent Instruction-Tuning Related Work

# Evaluation of Core Capabilities

- LLMs must show proficiency in a certain set of core tasks before they can generalize to all forms of user needs:
  - **Massive Multitask Language Understanding (MMLU)** (Hendrycks et al., 2021). 14079 questions covering 57 tasks including elementary mathematics, US history, business ethics, formal logic and so on.
  - **MATH** (Hendrycks et al., 2021) and **GSM8K** (Cobbe et al., 2021) two mathematical datasets.
  - **Big-Bench Hard (BBH)** (Suzgun et al., 2022) 23 challenging tasks, that were consistently proven to be difficult for LLMS to handle effectively.



# Holistic Evaluation of Language Models (HELM)

- (Liang et al., PMLR, 2023)
- An open-source framework and benchmark designed to provide transparent, multi-metric, and reproducible evaluations of large language models (LLMs).
- Multi-metric measurement
  - Unlike traditional benchmarks that may focus only on accuracy, HELM evaluates models on **multiple metrics across various use cases**, such as accuracy, robustness, efficiency, and so on.
- Broad coverage and recognition of incompleteness
  - Evaluates language models over a broad range of scenarios.
  - A holistic evaluation should provide a top-down taxonomy and make explicit all the major scenarios and metrics that are missing.
- Standardization
  - Each LM should be evaluated on the same scenarios to the extent possible.

# Topics for Today


Instruction Tuning

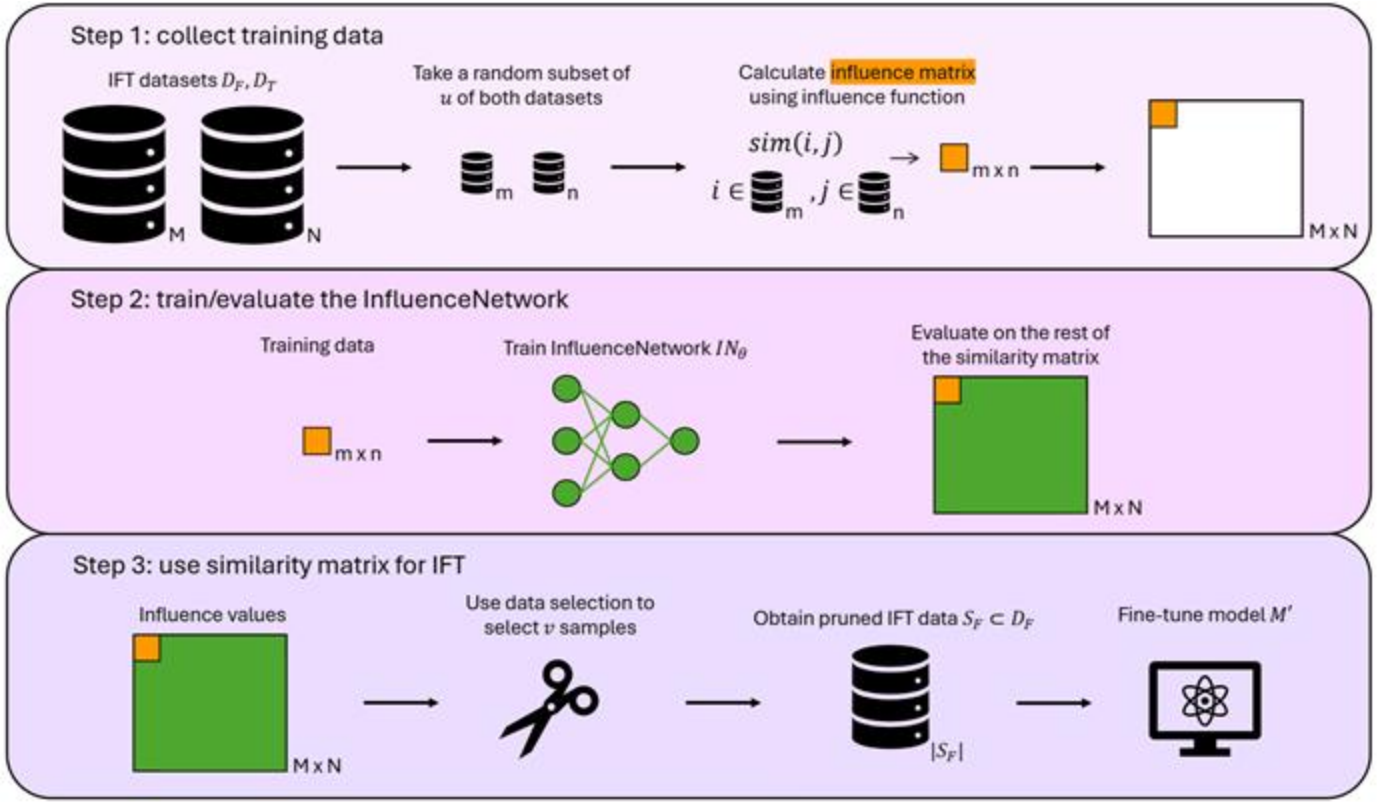
- Overview
- Instruction Tuning Datasets
- Evaluation
- Other Recent Instruction-Tuning Related Work

# Data Selection for Instruction Tuning

- LIMA has shown carefully designed, smaller subsets maybe sufficient. LIMA was manually designed, which is expensive.
- Selecting high-quality datasets to optimize instruction fine-tuning
- InstructMining (Cao et al., COLM, 2024): Utilizes natural language indicators (such as input length and perplexity) as a measure of data quality and applies them to evaluate unseen datasets.
- LESS (Xia et al., ICML, 2024):
  - constructs a highly reusable and transferable gradient datastore with low-dimensional gradient features
  - then selects examples based on their similarity to few-shot examples embodying a specific capability.
- DELIFT (Agarwal et al., ICLR, 2025):
  - uses a novel, computationally efficient utility metric (that uses conditional pointwise mutual information).
  - measures the informational value of each data sample by quantifying its effectiveness as an in-context example in improving model predictions for other samples.
  - reduces fine-tuning data requirements by up to 70% without compromising performance.

# NN-CIFT: Neural Networks for Efficient Instruction Fine-Tuning

Ishika Agarwal



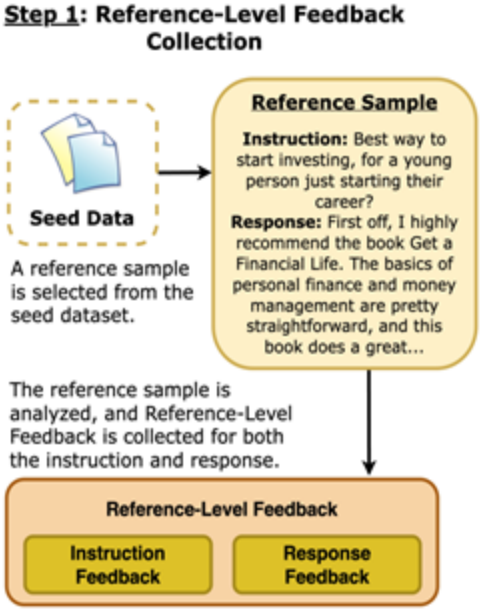

(Agarwal and Hakkani-Tür, NeurIPS 2025)

# Reference-Level Feedback

- (Mehri et al, 2025)
- **Reference samples** demonstrate desirable qualities
  - Using them as in-context example does not effectively capture these desirable qualities

- We propose **Reference-Level Feedback** to explicitly capture the desirable qualities from reference samples

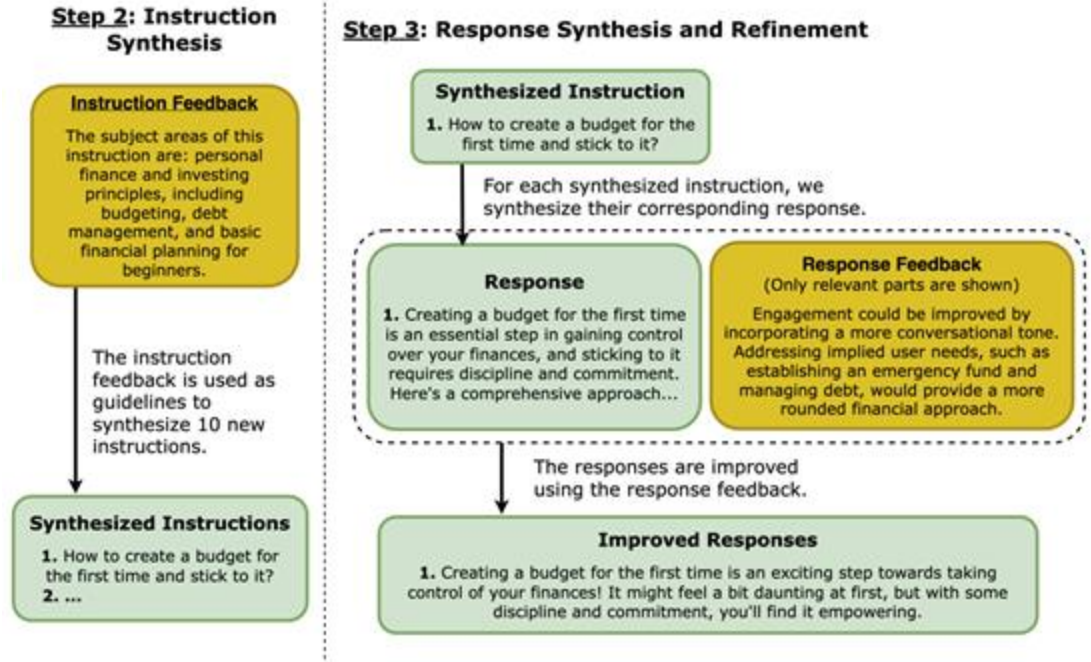
Shuhaib Mehri



# Reference-level Feedback

Shuhaib Mehri

- Then, we systematically propagate the desirable properties throughout the synthesis process



# Improving Instruction Following in LMs Through Activation Steering

- (Stolfo et al., ICLR, 2025)

- Enhance model adherence to constraints such as output format, length, and word inclusion, providing inference-time control over instruction following.

- Extract instruction specific vector representations from LMs and use them to steer models accordingly.

- Compositionality of activation steering: applying multiple instructions simultaneously.

# Improving Instruction Following in LMs Through Activation Steering (cont.)

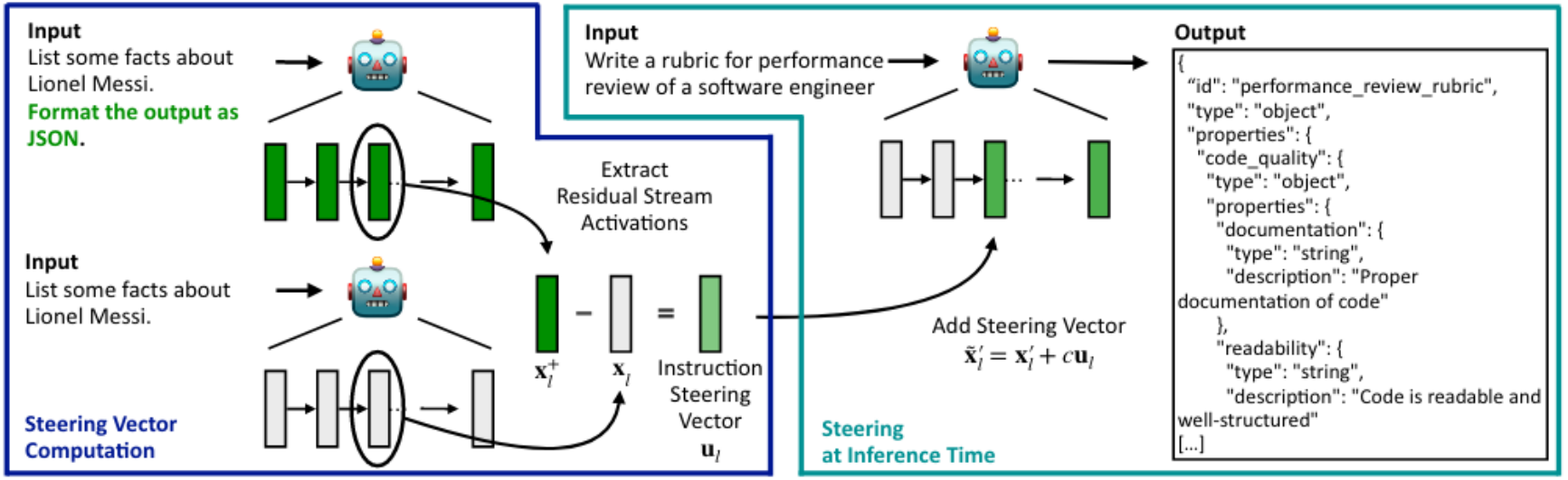



Figure 1: **Instruction Steering Process.** Steering vectors are computed as the difference in residual stream activations between inputs with and without the instruction. These vectors are then applied during inference to adjust the model's activations, guiding it to follow the desired instruction.

# Instruction Tuning – Benefits

- Bridges the gap between the LLM pre-training objective and what the real users want.

- Enables more control and predictable model behavior for the LLMs.
  - Allows the builders/developers to intervene with the model behavior.

- Is computationally efficient and can help LLMs quickly adapt their behavior to new domains and tasks without extensive training.

# Instruction Tuning – Challenges

- Creating high quality instructions and outputs that properly captures the desired behavior is non-trivial.

- Increasing concerns that, instruction tuning

    - improves tasks that are dominantly represented in the training datasets, for imitating models (Gudibande et al., 2023), and

    - mainly captures patterns and styles rather than truly learning the tasks (Kung and Peng, ACL, 2023)

# Next Week

Final Project Proposal Presentations

3-4 minute presentation by each team (no more than 3 slides!)

On Zoom

# Topic for the Week of October 13th

**Tuesday**

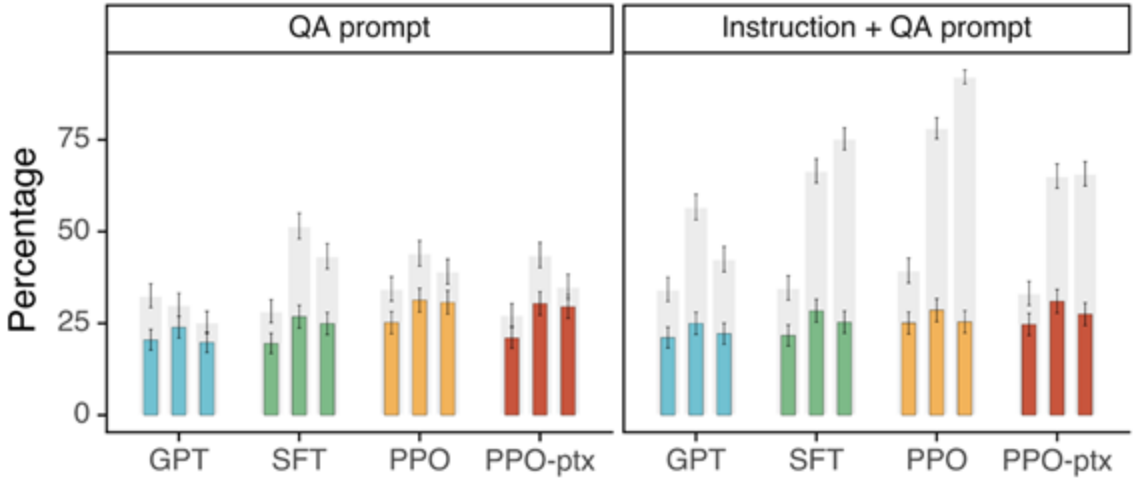
- Retrieval Augmented Generation

**Thursday**

- Tool Calling

# Evaluation: Results on public NLP datasets - TruthfulQA

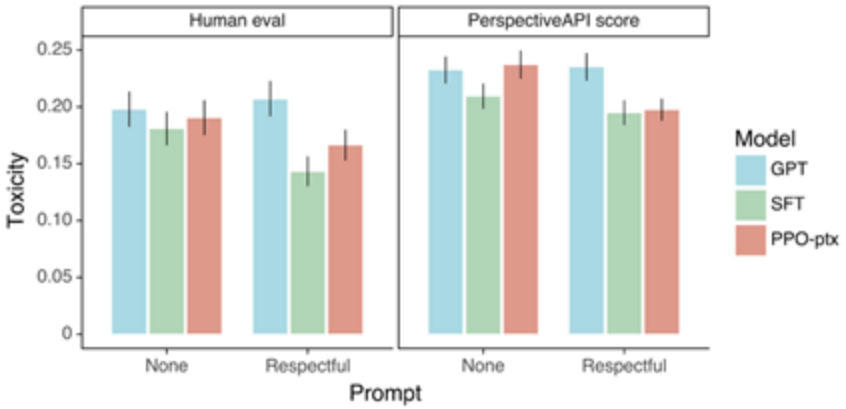
- TruthfulQA: measure whether a language model is truthful in generating answers to questions, consisting of 817 questions that span 38 categories, including health, law, finance, and politics
  - Metric: the percentage of its responses that a **human judges** to be true or informative
- **InstructGPT models show improvements in truthfulness over GPT-3.**




Gray bars indicate truthfulness and colored bars indicate truthfulness and informativeness.

# Evaluation: Results on public NLP datasets - RealToxicityPrompts

- RealToxicityPrompts: investigate the extent to which pre-trained LMs can be prompted to generate toxic language, such as racist and sexist, containing 100K prompts
  - Metric:
    - Automatic toxicity scores from Perspective API
    - Human evaluation of toxicity
- **InstructGPT shows small improvements in toxicity over GPT-3.**



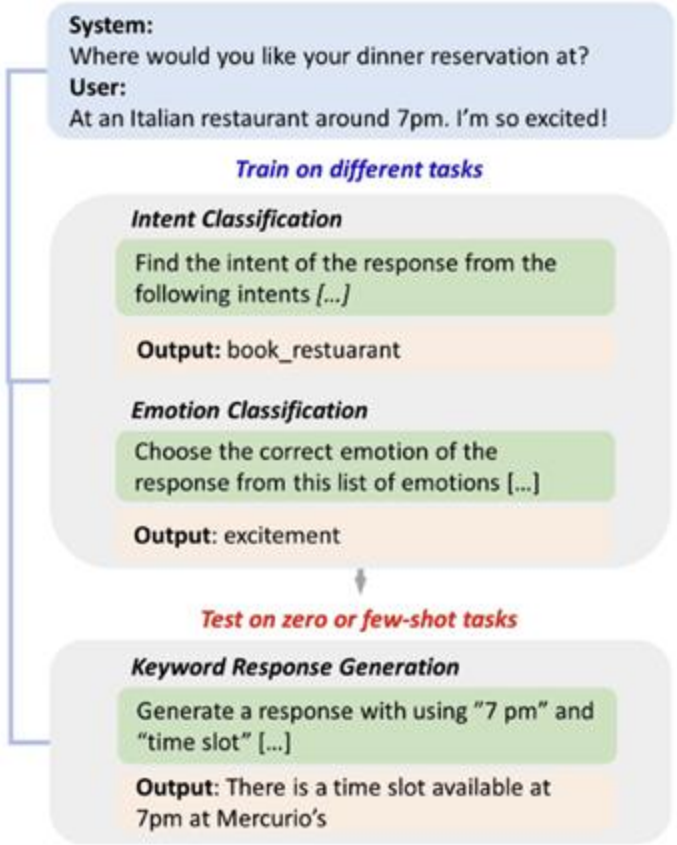
# Today's Papers

1. Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, Ryan Lowe. Training language models to follow instructions with human feedback. Arxiv, 2022.
2. Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, Hannaneh Hajishirzi. SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions. ACL, 2023.
3. Or Honovich, Thomas Scialom, Omer Levy, Timo Schick. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor. ACL, 2023.
4. Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. InstructDial: Improving Zero and Few-shot Generalization in Dialogue through Instruction Tuning. EMNLP, 2022.

# Introduction

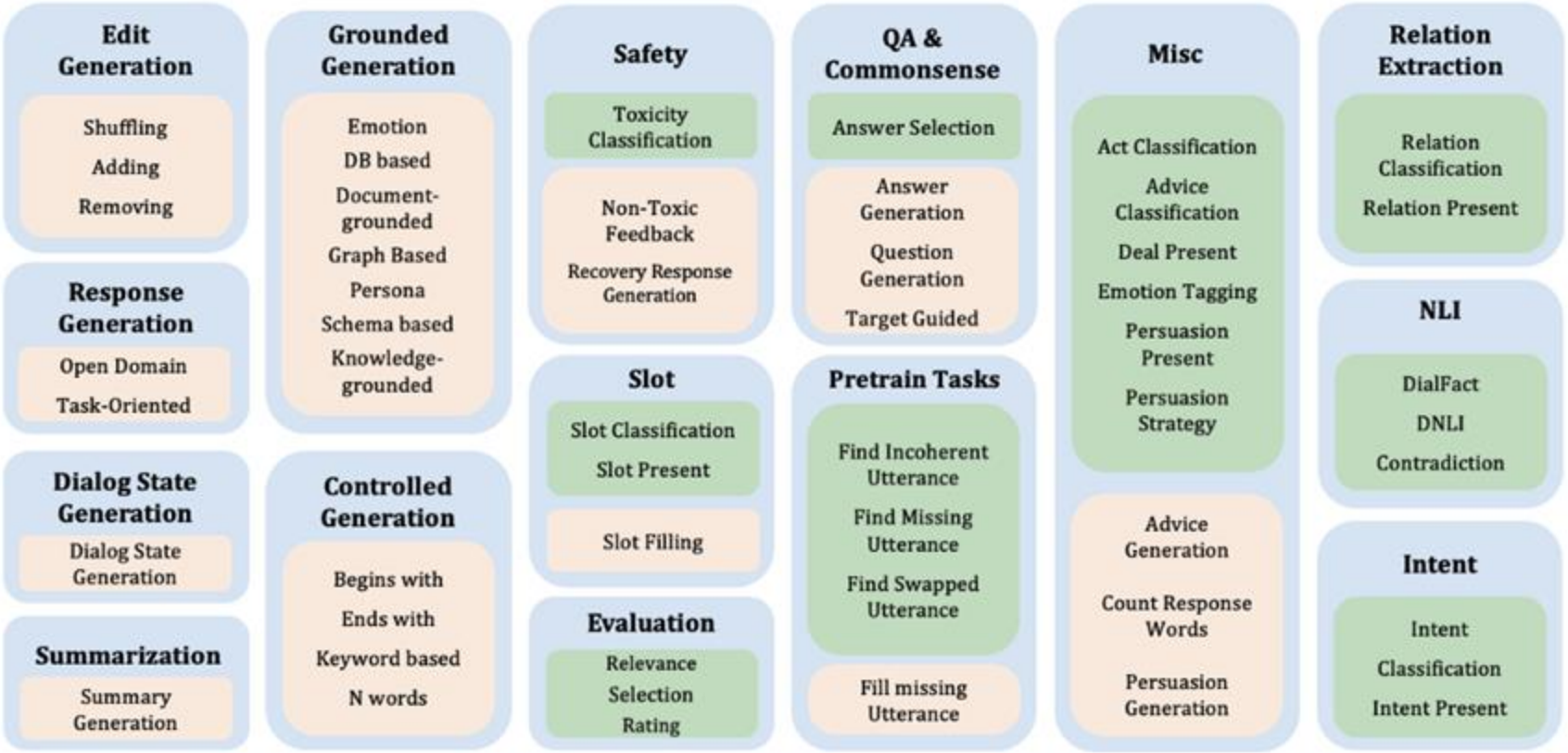
- Enhance zero-shot and few-shot performance
- on a variety of new tasks, such as evaluation and intent detection,
- by providing natural language instructions or prompts for dialogue related tasks during training.

- Instruction tuning for 48 dialogue tasks, 59 datasets
- open-sourced



# Motivation

- Previous works focus on general NLP tasks
- Few involve crowdsourced dialogue tasks
- Instruction tuning has not been systematically explored for dialogue-related tasks
- Enables developers and even non-expert users to leverage language models using natural language without large training datasets
- Can work for models that are significantly smaller than LLMs




INSTRUCTDIAL task taxonomy. **Green** represents classification and **orange** represents generation tasks.

# Contributions

- **InstructDial** - framework to systematically investigate instruction tuning for dialogue on a large collection of dialogue datasets and tasks
- Various analyses and establish baseline and upper bound performance for multiple tasks.
- Also provides integration of various task-specific dialogue metrics
- Introduces novel **meta-tasks** (e.g. select an instruction that matches with an input-output pair) to encourage models to adhere to the instructions.

# Background

- Cross-task setup: the model is tested on test instances of an unseen task in the training data.
- In instruction-tuning, model M is provided additional signal or meta-information about the task
  - Guides the model towards the expected output space of the new task.

# Task Schema and Formatting

- Prompts
- Task definitions
- Constraints
- Instance inputs / Examples
- Output

For each task, manually composed 3-10 task definitions and prompts.
For each instance, these are selected randomly.

# Task Schema and Formatting – Examples

# Meta Tasks

Introduced to help models learn associations between the instruction, the data and the task.

- **Instruction selection:** the model is asked to select the instruction corresponding to given input-output pair.
- **Instruction binary:** the model is asked to answer with yes or no on whether a given instruction corresponds to a given input-output pair.

For classification tasks, included a none-of-the-above (NOTA) option in examples, after removing the ground truth option.

# Baseline Models and InstructDial Models

- Encoder-decoder models
  - **T0-3B**
    - 3 billion parameter version of T5
  - **BART0**
    - 406 million parameters (8x smaller than T0-3B)
    - Based on BART-large
  - Both trained on a multi-task mixture of general non-dialogue tasks such as question answering, sentiment detection, and paraphrase identification
- **DIAL-BART0** and **DIAL-T0** are the result of the baseline models tuned on InstructDial

# Zero-shot Unseen Tasks Evaluation

Classification and generation tasks of varying complexity:

1. **Dialfact classification:** predict if an evidence supports, refutes, or does not have enough information to validate the response
2. **Relation classification:** predict the relation between two people in a dialogue
3. **Answer selection**: predict an answer to a conversational question
4. **Eval selection:** choose the most relevant response among the provided 4 option
5. **Knowledge grounded generation:** generate a response based on background knowledge
6. **Begins with generation:** generate a response that starts with the provided initial phrase

| Model | ES | AS | RC | DC | BW | | | | KG | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | ACC | ACC | ACC | ACC | B-2 | R-L | GR | F1 | B-2 | R-L | GR |
| *Baselines and Our Models* | | | | | | | | | | | | |
| BART0 | 22.2 | 58.5 | 6.3 | 33.7 | 4.2 | 4.9 | 12.0 | 45.7 | 17.4 | 5.3 | 13.3 | 23.9 |
| T0-3B | 45.9 | 60.2 | 1.3 | 33.1 | 14.1 | 4.1 | 10.7 | 55.5 | 14.2 | 3.2 | 10.7 | 78.0 |
| GPT-3 | 57.5 | 56.5 | 11.5 | 37.3 | 16.5 | 7.2 | 15.7 | 57.0 | 18.5 | 3.9 | 11.6 | **83.8** |
| DIAL-BART0 (Ours) | 66.7 | 59.5 | **17.8** | **35.6** | **56.3** | **13.1** | 26.4 | 60.2 | **27.8** | **11.1** | **21.4** | 68.5 |
| DIAL-T0 (Ours) | **74.4** | **65.2** | 6.4 | 34.5 | 55.0 | 12.4 | **26.5** | **61.3** | 22.2 | 7.2 | 16.5 | 69.8 |
| *Few and Full shot Variations* | | | | | | | | | | | | |
| DB-Few | 77.1 | 69.1 | 28.0 | 43.0 | 72.2 | 16.7 | 30.7 | 60.3 | 27.9 | 9.7 | 20.0 | 68.0 |
| DB-Full | 90.7 | 83.3 | 62.7 | 77.4 | 83.7 | 20.8 | 33.8 | 61.0 | 30.9 | 11.6 | 22.8 | 70.5 |
| *Model Ablations for DIAL-BART0* | | | | | | | | | | | | |
| DB-no-base | 40.1 | 52.7 | 17.1 | 35.1 | 53.9 | 12.0 | 26.6 | 57.8 | 29.8 | 12.0 | 22.8 | 69.6 |
| DB-no-instr | 23.0 | 43.2 | 15.1 | 35.4 | 50.0 | 13.0 | 27.0 | 61.1 | 30.1 | 11.2 | 20.8 | 65.7 |
| DB-no-nota | 66.5 | 57.2 | 17.2 | 35.9 | 56.1 | 10.9 | 25.3 | 58.4 | 28.0 | 11.0 | 21.4 | 67.6 |
| DB-no-meta | 44.5 | 52.0 | 14.1 | 35.4 | 52.5 | 14.1 | 28.1 | 61.3 | 29.6 | 11.8 | 22.1 | 70.5 |

- **BLEU-2 (B-2):** Measures the overlap of n-grams between machine-generated text and reference text.
- **RougeL (R-L):** Evaluates the quality of summaries or translations by comparing the longest common subsequences between them.
- **GRADE (GR)**: Assesses the coherence and quality of generated responses by considering response length, repetition, and informativeness.

| Model | ES | AS | RC | DC | BW | | | | KG | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | ACC | ACC | ACC | ACC | B-2 | R-L | GR | F1 | B-2 | R-L | GR |
| *Baselines and Our Models* | | | | | | | | | | | | |
| BART0 | 22.2 | 58.5 | 6.3 | 33.7 | 4.2 | 4.9 | 12.0 | 45.7 | 17.4 | 5.3 | 13.3 | 23.9 |
| T0-3B | 45.9 | 60.2 | 1.3 | 33.1 | 14.1 | 4.1 | 10.7 | 55.5 | 14.2 | 3.2 | 10.7 | 78.0 |
| GPT-3 | 57.5 | 56.5 | 11.5 | 37.3 | 16.5 | 7.2 | 15.7 | 57.0 | 18.5 | 3.9 | 11.6 | **83.8** |
| DIAL-BART0 (Ours) | 66.7 | 59.5 | **17.8** | **35.6** | **56.3** | **13.1** | 26.4 | 60.2 | **27.8** | **11.1** | **21.4** | 68.5 |
| DIAL-T0 (Ours) | **74.4** | **65.2** | 6.4 | 34.5 | 55.0 | 12.4 | **26.5** | **61.3** | 22.2 | 7.2 | 16.5 | 69.8 |
| *Few and Full shot Variations* | | | | | | | | | | | | |
| DB-Few | 77.1 | 69.1 | 28.0 | 43.0 | 72.2 | 16.7 | 30.7 | 60.3 | 27.9 | 9.7 | 20.0 | 68.0 |
| DB-Full | 90.7 | 83.3 | 62.7 | 77.4 | 83.7 | 20.8 | 33.8 | 61.0 | 30.9 | 11.6 | 22.8 | 70.5 |
| *Model Ablations for DIAL-BART0* | | | | | | | | | | | | |
| DB-no-base | 40.1 | 52.7 | 17.1 | 35.1 | 53.9 | 12.0 | 26.6 | 57.8 | 29.8 | 12.0 | 22.8 | 69.6 |
| DB-no-instr | 23.0 | 43.2 | 15.1 | 35.4 | 50.0 | 13.0 | 27.0 | 61.1 | 30.1 | 11.2 | 20.8 | 65.7 |
| DB-no-nota | 66.5 | 57.2 | 17.2 | 35.9 | 56.1 | 10.9 | 25.3 | 58.4 | 28.0 | 11.0 | 21.4 | 67.6 |
| DB-no-meta | 44.5 | 52.0 | 14.1 | 35.4 | 52.5 | 14.1 | 28.1 | 61.3 | 29.6 | 11.8 | 22.1 | 70.5 |

**Instruction tuning on INSTRUCTDIAL improves performance on unseen dialogue tasks**

- Eval selection, Relation classification, and Begins with perform 3x better than baseline
- Significantly better than GPT, besides Dialfact classification

| Model | ES | AS | RC | DC | BW | | | | KG | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | ACC | ACC | ACC | ACC | B-2 | R-L | GR | F1 | B-2 | R-L | GR |
| *Baselines and Our Models* | | | | | | | | | | | | |
| BART0 | 22.2 | 58.5 | 6.3 | 33.7 | 4.2 | 4.9 | 12.0 | 45.7 | 17.4 | 5.3 | 13.3 | 23.9 |
| T0-3B | 45.9 | 60.2 | 1.3 | 33.1 | 14.1 | 4.1 | 10.7 | 55.5 | 14.2 | 3.2 | 10.7 | 78.0 |
| GPT-3 | 57.5 | 56.5 | 11.5 | 37.3 | 16.5 | 7.2 | 15.7 | 57.0 | 18.5 | 3.9 | 11.6 | **83.8** |
| DIAL-BART0 (Ours) | 66.7 | 59.5 | **17.8** | **35.6** | **56.3** | **13.1** | 26.4 | 60.2 | **27.8** | **11.1** | **21.4** | 68.5 |
| DIAL-T0 (Ours) | **74.4** | **65.2** | 6.4 | 34.5 | 55.0 | 12.4 | **26.5** | **61.3** | 22.2 | 7.2 | 16.5 | 69.8 |
| *Few and Full shot Variations* | | | | | | | | | | | | |
| DB-Few | 77.1 | 69.1 | 28.0 | 43.0 | 72.2 | 16.7 | 30.7 | 60.3 | 27.9 | 9.7 | 20.0 | 68.0 |
| DB-Full | 90.7 | 83.3 | 62.7 | 77.4 | 83.7 | 20.8 | 33.8 | 61.0 | 30.9 | 11.6 | 22.8 | 70.5 |
| *Model Ablations for DIAL-BART0* | | | | | | | | | | | | |
| DB-no-base | 40.1 | 52.7 | 17.1 | 35.1 | 53.9 | 12.0 | 26.6 | 57.8 | 29.8 | 12.0 | 22.8 | 69.6 |
| DB-no-instr | 23.0 | 43.2 | 15.1 | 35.4 | 50.0 | 13.0 | 27.0 | 61.1 | 30.1 | 11.2 | 20.8 | 65.7 |
| DB-no-nota | 66.5 | 57.2 | 17.2 | 35.9 | 56.1 | 10.9 | 25.3 | 58.4 | 28.0 | 11.0 | 21.4 | 67.6 |
| DB-no-meta | 44.5 | 52.0 | 14.1 | 35.4 | 52.5 | 14.1 | 28.1 | 61.3 | 29.6 | 11.8 | 22.1 | 70.5 |

**Larger models are not necessarily better across tasks**

- T0-3B and DIAL-T0 perform better on the Eval selection and Answer Selection
- DIAL-T0 and DIAL-BART0 perform better on the rest of the unseen task

| Model | ES | AS | RC | DC | BW | | | | KG | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | ACC | ACC | ACC | ACC | B-2 | R-L | GR | F1 | B-2 | R-L | GR |
| *Baselines and Our Models* | | | | | | | | | | | | |
| BART0 | 22.2 | 58.5 | 6.3 | 33.7 | 4.2 | 4.9 | 12.0 | 45.7 | 17.4 | 5.3 | 13.3 | 23.9 |
| T0-3B | 45.9 | 60.2 | 1.3 | 33.1 | 14.1 | 4.1 | 10.7 | 55.5 | 14.2 | 3.2 | 10.7 | 78.0 |
| GPT-3 | 57.5 | 56.5 | 11.5 | 37.3 | 16.5 | 7.2 | 15.7 | 57.0 | 18.5 | 3.9 | 11.6 | **83.8** |
| DIAL-BART0 (Ours) | 66.7 | 59.5 | **17.8** | **35.6** | **56.3** | **13.1** | 26.4 | 60.2 | **27.8** | **11.1** | **21.4** | 68.5 |
| DIAL-T0 (Ours) | **74.4** | **65.2** | 6.4 | 34.5 | 55.0 | 12.4 | **26.5** | **61.3** | 22.2 | 7.2 | 16.5 | 69.8 |
| *Few and Full shot Variations* | | | | | | | | | | | | |
| DB-Few | 77.1 | 69.1 | 28.0 | 43.0 | 72.2 | 16.7 | 30.7 | 60.3 | 27.9 | 9.7 | 20.0 | 68.0 |
| DB-Full | 90.7 | 83.3 | 62.7 | 77.4 | 83.7 | 20.8 | 33.8 | 61.0 | 30.9 | 11.6 | 22.8 | 70.5 |
| *Model Ablations for DIAL-BART0* | | | | | | | | | | | | |
| DB-no-base | 40.1 | 52.7 | 17.1 | 35.1 | 53.9 | 12.0 | 26.6 | 57.8 | 29.8 | 12.0 | 22.8 | 69.6 |
| DB-no-instr | 23.0 | 43.2 | 15.1 | 35.4 | 50.0 | 13.0 | 27.0 | 61.1 | 30.1 | 11.2 | 20.8 | 65.7 |
| DB-no-nota | 66.5 | 57.2 | 17.2 | 35.9 | 56.1 | 10.9 | 25.3 | 58.4 | 28.0 | 11.0 | 21.4 | 67.6 |
| DB-no-meta | 44.5 | 52.0 | 14.1 | 35.4 | 52.5 | 14.1 | 28.1 | 61.3 | 29.6 | 11.8 | 22.1 | 70.5 |

**Few-shot (DB-Few) training significantly improves performance (100 examples/task)**

- 12-16% improvements on the Eval selection, Answer selection, and Dialfact classification tasks
- 30-50% improvement on the Begins with and Relation classification tasks

**Full-shot (DB-Full) improves across all test tasks (5000 examples/task)**

| Model | ES | AS | RC | DC | BW | | | | KG | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | ACC | ACC | ACC | ACC | B-2 | R-L | GR | F1 | B-2 | R-L | GR |
| *Baselines and Our Models* | | | | | | | | | | | | |
| BART0 | 22.2 | 58.5 | 6.3 | 33.7 | 4.2 | 4.9 | 12.0 | 45.7 | 17.4 | 5.3 | 13.3 | 23.9 |
| T0-3B | 45.9 | 60.2 | 1.3 | 33.1 | 14.1 | 4.1 | 10.7 | 55.5 | 14.2 | 3.2 | 10.7 | 78.0 |
| GPT-3 | 57.5 | 56.5 | 11.5 | 37.3 | 16.5 | 7.2 | 15.7 | 57.0 | 18.5 | 3.9 | 11.6 | **83.8** |
| DIAL-BART0 (Ours) | 66.7 | 59.5 | **17.8** | **35.6** | **56.3** | **13.1** | 26.4 | 60.2 | **27.8** | **11.1** | **21.4** | 68.5 |
| DIAL-T0 (Ours) | **74.4** | **65.2** | 6.4 | 34.5 | 55.0 | 12.4 | **26.5** | **61.3** | 22.2 | 7.2 | 16.5 | 69.8 |
| *Few and Full shot Variations* | | | | | | | | | | | | |
| DB-Few | 77.1 | 69.1 | 28.0 | 43.0 | 72.2 | 16.7 | 30.7 | 60.3 | 27.9 | 9.7 | 20.0 | 68.0 |
| DB-Full | 90.7 | 83.3 | 62.7 | 77.4 | 83.7 | 20.8 | 33.8 | 61.0 | 30.9 | 11.6 | 22.8 | 70.5 |
| *Model Ablations for DIAL-BART0* | | | | | | | | | | | | |
| DB-no-base | 40.1 | 52.7 | 17.1 | 35.1 | 53.9 | 12.0 | 26.6 | 57.8 | 29.8 | 12.0 | 22.8 | 69.6 |
| DB-no-instr | 23.0 | 43.2 | 15.1 | 35.4 | 50.0 | 13.0 | 27.0 | 61.1 | 30.1 | 11.2 | 20.8 | 65.7 |
| DB-no-nota | 66.5 | 57.2 | 17.2 | 35.9 | 56.1 | 10.9 | 25.3 | 58.4 | 28.0 | 11.0 | 21.4 | 67.6 |
| DB-no-meta | 44.5 | 52.0 | 14.1 | 35.4 | 52.5 | 14.1 | 28.1 | 61.3 | 29.6 | 11.8 | 22.1 | 70.5 |

## Meta tasks and NONE OF THE ABOVE

- Large performance drop in ***DB-no-meta***
- Slight performance drop in ***DB-no-nota***

## Instructions

- ***DB-no-instr*** has worse performance than *DIAL-BART0*

# Zero-shot and Few-shot Dialogue Tasks - Intent Prediction

- Predict intent of a given utterance (Banking77)
- Competitive performance in few-shot setting
  - Better than *PPTOD* (2x parameters)
- *BART0* struggles in zero-shot setting

| Model | Accuracy |
|---|---|
| ConvERT (Casanueva et al., 2020) | 83.32 |
| ConvERT + USE (Casanueva et al., 2020) | 85.19 |
| Example-Driven (Mehri and Eric, 2021) | 85.95 |
| $\text{PPTOD}_{base}$ (Su et al., 2022b) | 82.81 |
| $\text{PPTOD}_{large}$ (Su et al., 2022b) | 84.12 |
| DIAL-BART0 (Ours) | 84.30 |
| BART0 (zero-shot) | 14.72 |
| DIAL-BART0 (Ours, zero-shot) | 58.02 |

# Zero-shot and Few-shot Dialogue Tasks - Slot Filling

- Detecting slot values in a given utterance (Restaurants8k & DSTC8)
- 36.9 point improvement in zero-shot slot filling
- Significant improvement in few-shot setting

| Model | F1 |
|---|---|
| ConVEx (Henderson and Vulić, 2020) | 5.2 |
| Coach+TR (Liu et al., 2020) | 10.7 |
| GenSF (Mehri and Eskenazi, 2021) | 19.5 |
| DIAL-BART0 (Ours) | 56.4 |

Table 4: Zero-shot slot filling results on the Restaurant8k corpus.

| Domain | GenSF | DIAL-BART0 (Ours) |
|---|---|---|
| Buses | 90.5 | 97.8 |
| Events | 91.2 | 94.3 |
| Homes | 93.7 | 96.5 |
| Rental Cars | 86.7 | 94.2 |

Table 5: Few-shot slot filling F1 scores on DSTC8 data.

# Zero-shot and Few-shot Dialogue Tasks - Dialogue State Tracking

- Similar to PPTOD, first pre-trained on 7 datasets
  - KVRET
  - WOZ
  - CamRest676
  - MSR-E2E
  - Frames
  - TaskMaster
  - Schema-Guided Dialogue
- Then trained on 1% and 5% splits of MultiWOZ 2.0

- Competitive performance (*PPTOD* has 2x parameters)

| Model | 1% data | 5% data |
|---|---|---|
| $\text{PPTOD}_{base}$ | 29.7 | 40.2 |
| DIAL-BART0 (Ours) | 29.2 | 38.1 |

Table 6: Joint goal accuracy for dialogue state tracking in few-shot setting on 1% and 5% data of Multiwoz.

# Limitations

- The instructions and prompts used in the study are not crowdsourced, potentially limiting language diversity and quantity.
- Room for improvement in zero-shot setting (compared to few-shot and full-shot)
- Instances of task interference were observed, indicating the need for methods to mitigate negative effects and prevent task forgetting.

# Conclusions

- InstructDial addresses a critical void by systematically exploring instruction tuning for dialogue tasks, which has been largely overlooked in previous research focusing on general NLP tasks.
- Notable achievements in both zero-shot and few-shot performance across various dialogue tasks, including intent prediction and slot filling.
- The introduction of novel meta-tasks incentivizes models to adhere to instructions, contributing to the enhancement of overall system effectiveness and usability.

# Core Dataset Generation – Example Inputs/Output

**Example 1**

**Instruction:** You are given a science question (easy-level) and four answer options (associated with "A", "B", "C", "D"). Your task is to find the correct answer based on scientific facts, knowledge, and reasoning. Do not generate anything else apart from one of the following characters: 'A', 'B, 'C', 'D'. There is only one correct answer for each question.

**Input:** Which part of a bicycle BEST moves in a circle? (A) Seat (B) Frame (C) Foot pedal (D) Kickstand

**Constraints:** The output should be one of the following characters: 'A', 'B, 'C', 'D'.

**Example 2**

**Instruction:** You are given a negative review and your task is to convert it to a positive review by one or more making minimal changes. Avoid changing the context of the review.

**Input:** we stood there in shock, because we never expected this.

**Constraints:** None.

**Example 3**

**Instruction:** In this task, you are given two sentences taken from a conversation, and your job is to classify whether these given sentences are sequential or not. We will mark the given sentence pair as 'True' if it's sequential, otherwise 'False'. The two sentences are spoken by two different people.

**Input:** Noah: When and where are we meeting? :), Madison: I thought you were busy...?

**Constraints:** None.

**Example 4**

**Instruction:** In this task, you will be given a profile of someone and your job is to generate a set of interesting questions that can lead to a conversation with the person.

**Input:** Yvonne has been playing the violin since she was four years old. She loves all kinds of music, but her favorite composer is Bach.

**Constraints:** None.

# Template Expansion (cont.)

**Example 1**

**Instruction:** In this task, you are given an article. Your task is to summarize the article in a sentence.

**Input:** {INPUT}

**Alternative formulation:** My college roommate asked me what this article means: "{INPUT}". So I recapped it in layman's terms:

**Example 2**

**Instruction:** This task is about writing a correct answer for the reading comprehension task. Based on the information provided in a given passage…

**Input:** {INPUT}

**Alternative formulation:** {INPUT} Based on the given context, the answer to the question is

**Example 3**

**Instruction:** In this task, you are asked to determine whether the given recipe is for a savory or sweet dish. If it is for a savory dish, output "SAVORY". If the recipe is for a sweet dish, output "SWEET".

**Input:** {INPUT}

**Alternative formulation:** Given the following recipe, {INPUT}, is the dish savory or sweet? Your output should be "SAVORY" or "SWEET"

Usually shorter and less formal.

# Resulting Dataset

- Core dataset generation
  - 58K examples
- Reformulation
  - 240K examples

| Instruction | Category |
|---|---|
| You need to answer the question 'Is this a good experiment design?', given an experiment scenario. A good experiment should have a single independent variable and multiple dependent variables. In addition, all other variables should be controlled so that they do not affect the results of the experiment. | Experiment Verification |
| You are given a recipe for baking muffins that contains some errors. Your task is to correct the errors in the instructions by replacing each underlined word with the correct one from the options provided. | Recipe Correction |
| You will be given a piece of text that contains characters, places, and objects. For each character in the text, you need to determine whether they are static or dynamic. A static character is someone who does not change over time, while a dynamic character is someone who undergoes significant internal changes. | Character Categorization |
| In this task, you are asked to generate a limerick given two rhyming words. A limerick is a five-line poem with the following rhyme scheme: AABBA. The first, second and fifth lines must be of three beats, while the third and fourth lines must be of two beats each. Additionally, all poems should have the same meter (e.g., iambic pentameter) | Poem Generation |
| I'm not sure what this idiom means: "{INPUT}". Could you give me an example? | Idiom Explanation |
| {INPUT} By analyzing the writing styles of the two passages, do you think they were written by the same author? | Author Classification |
| I need to invent a new word by combining parts of the following words: {INPUT}. In what order should I put the parts together? | Word Invention |
| What is the punchline to the following joke? {INPUT} | Humor Understanding |