Instruction Tuning

Large Language Models: Introduction and Recent Advances

ELL881 · AIL821

Gaurav Pandey
Research Scientist, IBM Research

How to make ChatGPT ?

- Pre-Training
  - This is the point where most of the reasoning power is infused in the model.
  - Data – Billions of tokens of unstructured text from the internet
- Instruction Tuning
  - Trains models to follow natural language instructions
  - Data – Several thousand (Task/Instruction, Output) examples
- Reinforcement Learning from Human Feedback
  - Show the output(s) generated by models to humans/reward model
  - Collect feedback in the form of preferences.
  - Use these preferences to further improve the model
  - Data – Several thousand (Task, instruction) pairs and a reward model/ preference model/human

How to make ChatGPT ?

- Pre-Training
  - This is the point where most of the reasoning power is infused in the model.
  - Data – Billions of tokens of unstructured text from the internet
- Instruction Tuning
  - Trains models to follow natural language instructions
  - Data – Several thousand (Task/Instruction, Output) examples
- Reinforcement Learning from Human Feedback
  - Show the output(s) generated by models to humans/reward model
  - Collect feedback in the form of preferences.
  - Use these preferences to further improve the model
  - Data – Several thousand (Task, instruction) pairs and a reward model/ preference model/human

Table of Contents

Why Do We Need Instruction Training?

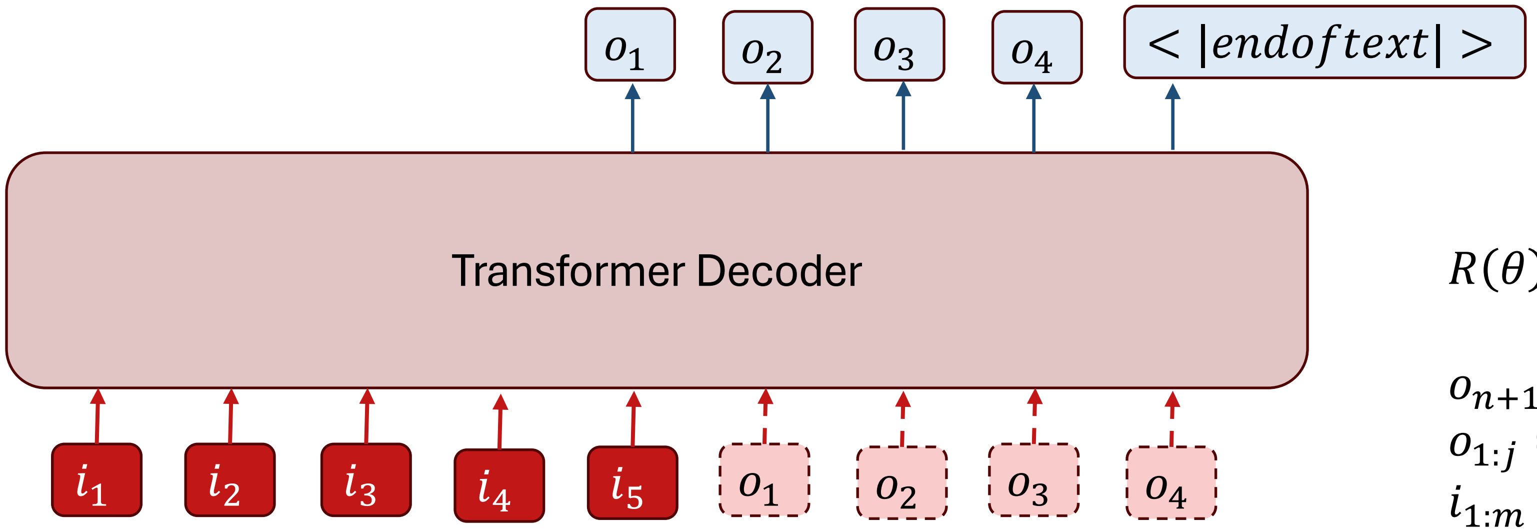
To bridge the gap between	Observed behavior: Next word prediction Desired Behavior: Instruction Following
To allow behavior modification during inference	Meta-instruction: Answer all questions as William Shakespeare would.
Catch	The instruction-tuning data should be diverse and have high coverage

Content Credit: Instruction Tuning for Large Language Models: A Survey

Training Loss

How to train? (Decoder-only models)

- Given (instruction, output) pairs
  - Tokenized $instruction = (i_1, \dots, i_m)$ $output = (o_1, \dots, o_n)$



$$R(\theta) = \sum_{j=0}^{n} \log p_\theta(o_{j+1} | o_{1:j}, i_{1:m})$$

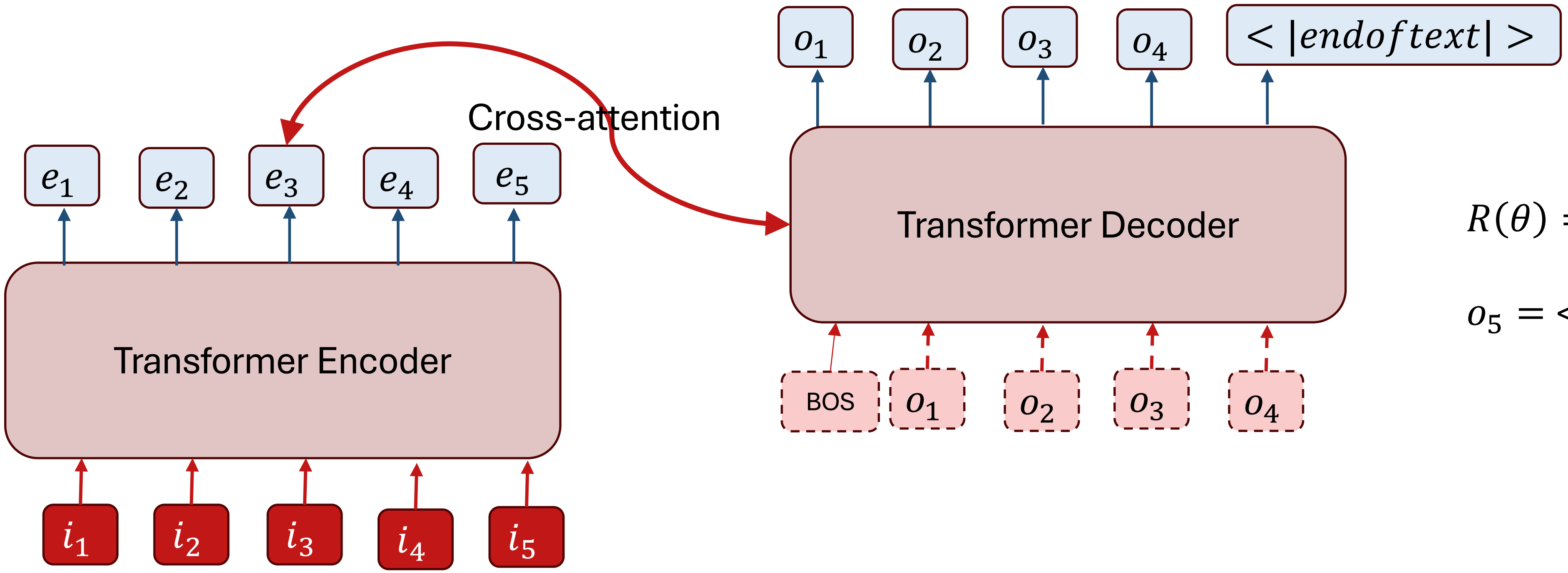
$$o_{n+1} = < |endoftext| >$$

$$o_{1:j} = o_1, \dots, o_j$$

$$i_{1:m} = i_1, \dots, i_m$$

How to train? (Encoder-Decoder Models)

- Given (instruction, output) pairs
  - Tokenized $instruction = (i_1, \ldots, i_m)$ $output = (o_1, \ldots, o_n)$



$$R(\theta) = \sum_{j=0}^{n} \log p_\theta(o_{j+1} | o_{1:j}, i_{1:m})$$

$$o_5 = < |endoftext| >$$

Getting the Data

Where does the data come from?

- Human-crafted
  - Flan-2021
    - Transforms NLP benchmarks into natural language input-output pairs.

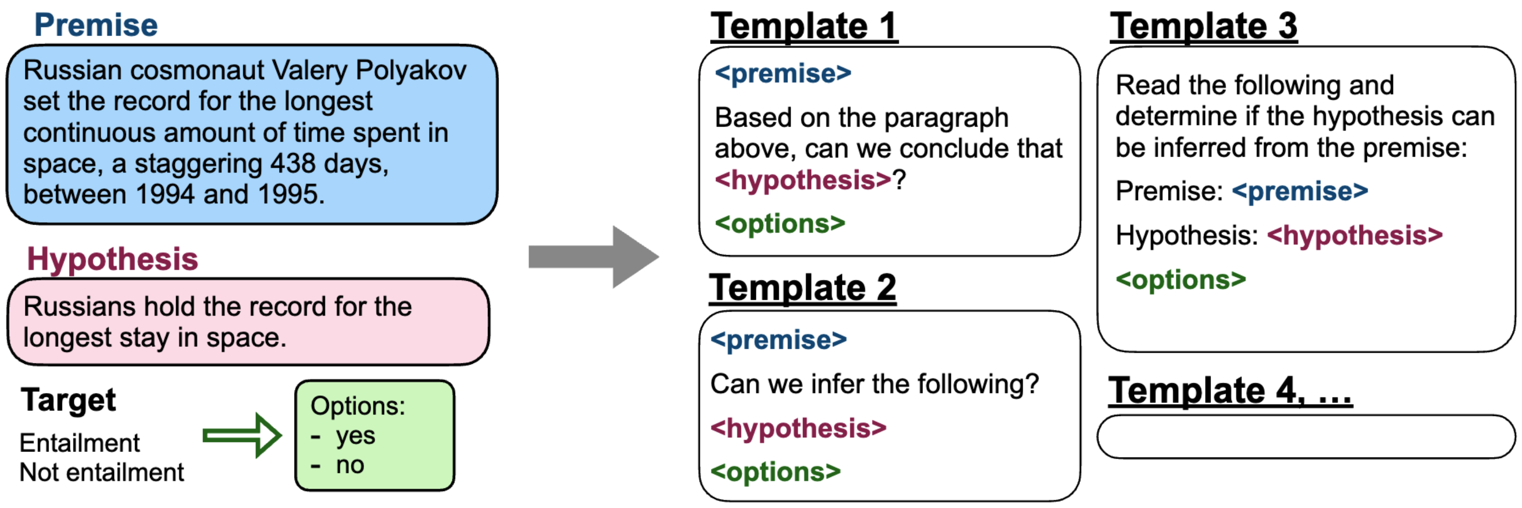
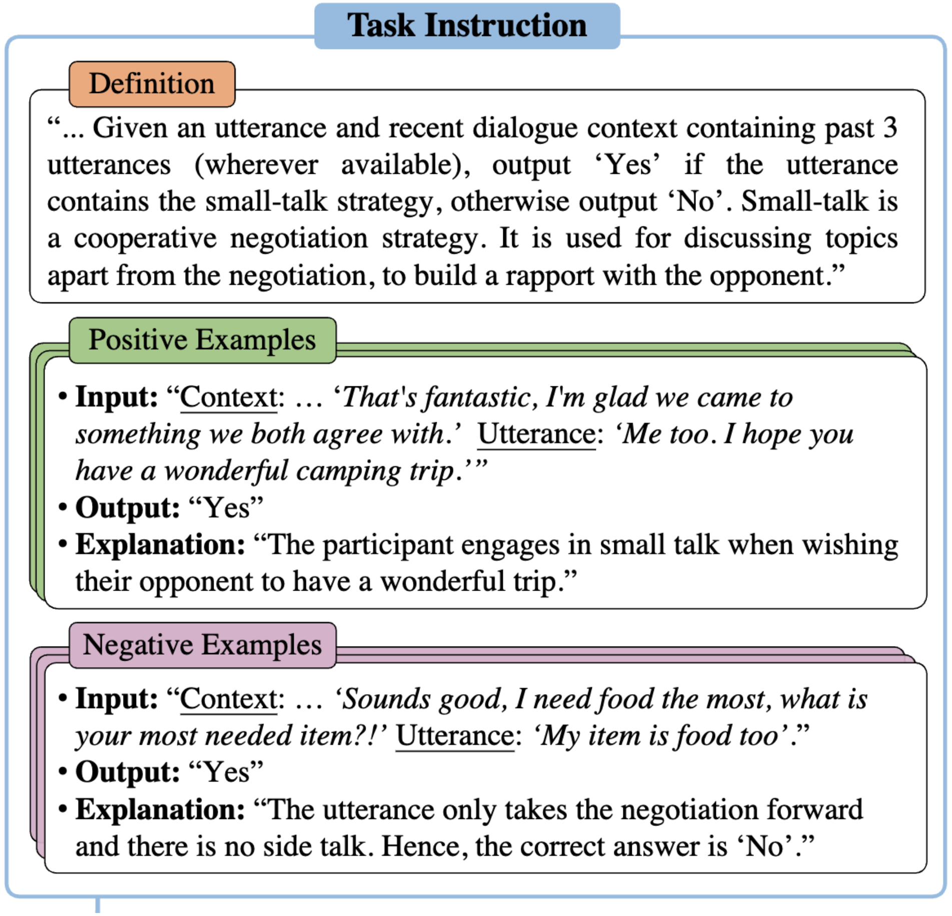


Figure 4: Multiple instruction templates describing a natural language inference task.

Credit: The Flan Collection: Designing Data and Methods for Effective Instruction Tuning

SuperNatural Instructions



Tasks contributed by NLP practitioners

Creative modification of existing NLP tasks

Synthetic tasks that can be communicated in few sentences

Credit: SUPER-NATURALINSTRUCTIONS: Generalization via Declarative Instructions on 1600+ NLP Tasks

Synthetic Instruction-Tuning Data

Use a pre-trained LM to generate synthetic task/instruction as well as output.

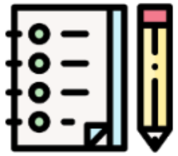
- Cheap and easy to obtain
- Often better quality than human-crafted data.

We will look at 4 popular approaches for synthetic data generation for instruction tuning:

- Self-Instruct
- Evol-Instruct
- Orca
- Instruction Back-translation

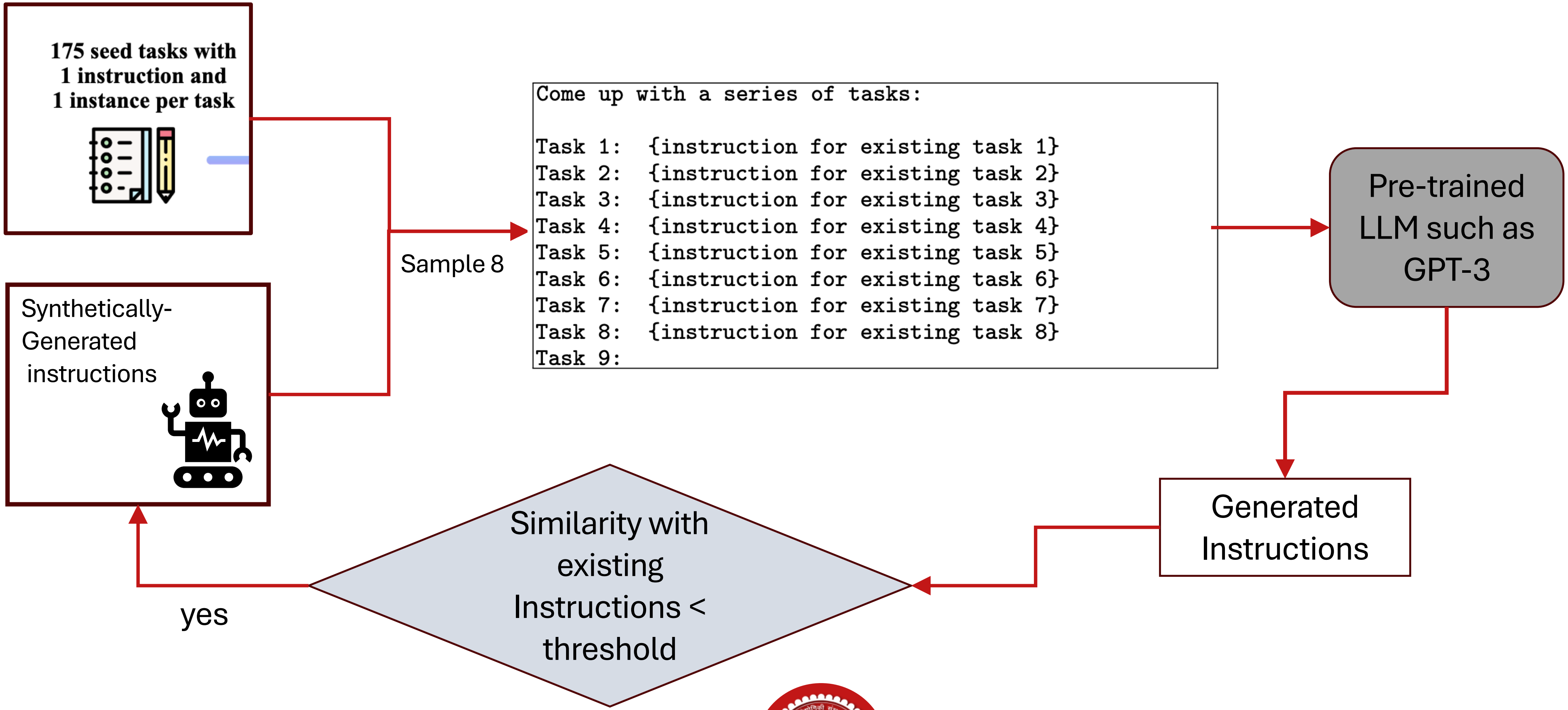
Self-Instruct

- Given: 175 seed tasks with 1 instruction and 1 instance per task



- Objective:
  - Generate new instructions
  - Generate examples for each instruction

The Self-Instruct Process – Instruction Generation



The Self-Instruct Process – 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: {instruction for the target task}
Is it classification?

The Self-Instruct Process – Instance Generation

- Given an instruction, generate instances that follow the instruction.
- In-context learning can be used to generate instances for an instruction
- Input-First

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

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

Task: {Instruction for the target task}

The Self-Instruct Process – Instance Generation - II

Output First

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

Task: Classify the sentiment of the sentence into positive, negative, or mixed.
Class label: mixed
Sentence: I enjoy the flavor of the restaurant but their service is too slow.
Class label: Positive
Sentence: I had a great day today. The weather was beautiful and I spent time with friends.
Class label: Negative

Task: {instruction for the target task}

Self-Instruct: The complete pipeline

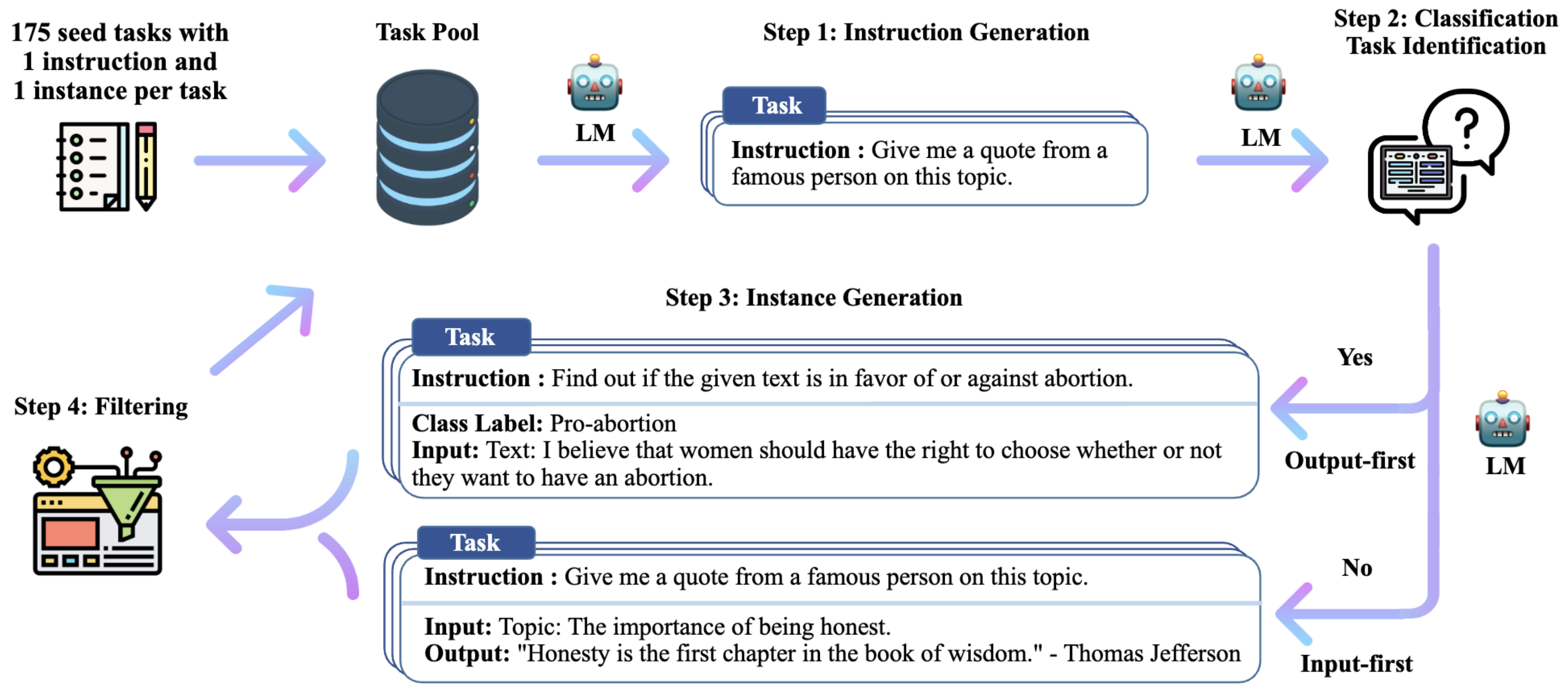


Image Credit: SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions

Evaluation results on unseen tasks from SUPERNI

Model	# Params	ROUGE-L
GPT3	175B	6.8
InstructGPT001	175B	40.8
GPT3 + T0 Training	175B	37.9
GPT3 SELF-INST	175B	39.9
GPT3 + SUPERNI Training	175B	49.5
GPT3 SELF-INST + SUPERNI Training	175B	51.6

Human evaluation on 252 instructions

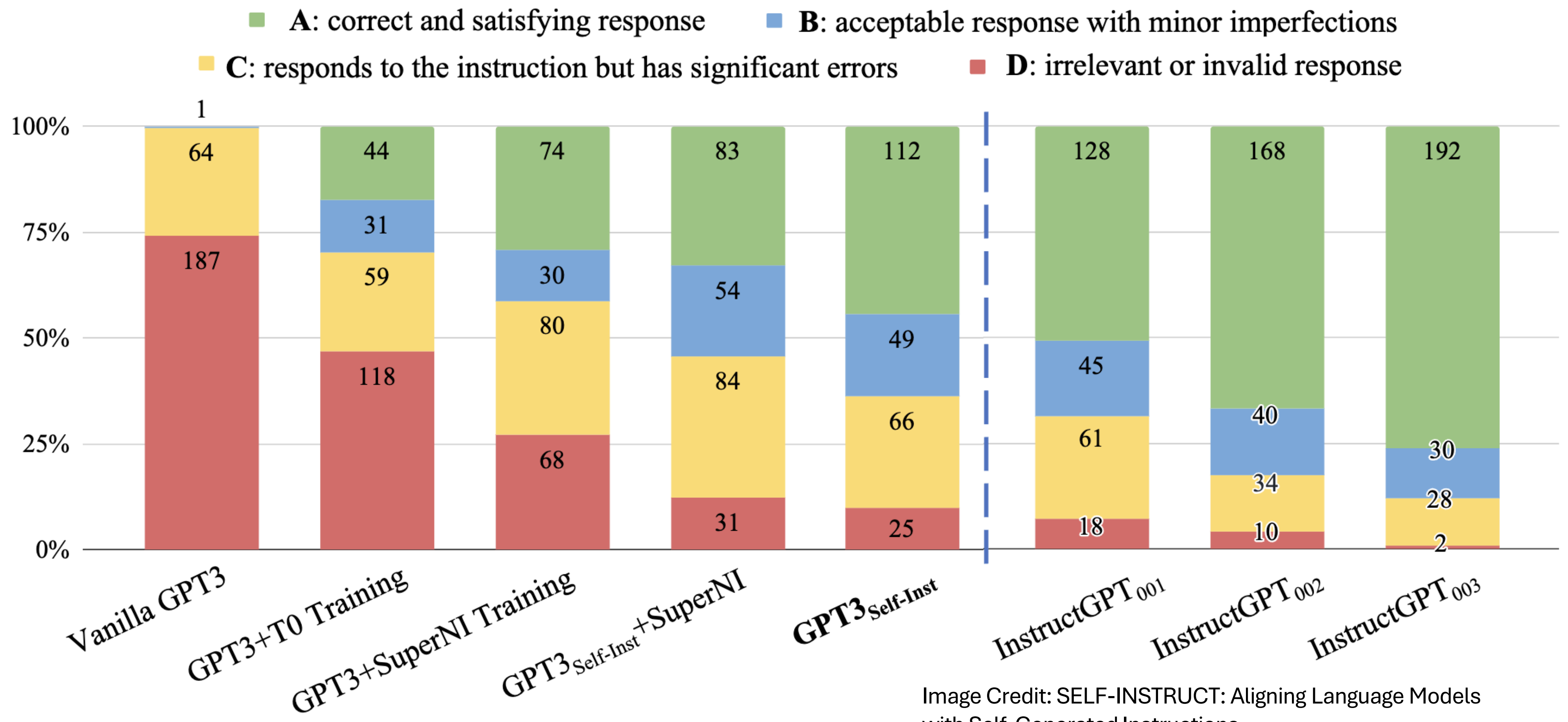


Image Credit: SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions

Evol-Instruct

Motivation:

Most of the instruction datasets contain only simple instructions.

LLMs can be used to make instructions more complex.

Instruction Evolver

An LLM that uses prompts to evolve instructions.

Instruction Eliminator

Checks whether the evolution fails.

- Non-informative responses

Instruction Evolver – In-Depth Evolution

- Add constraints
- Deepening
- Concretizing
- Increase Reasoning

I want you act as a Prompt Rewriter.
Your objective is to rewrite a given prompt into a more complex version to make those famous AI systems
(e.g., ChatGPT and GPT4) a bit harder to handle.
But the rewritten prompt must be reasonable and must be understood and responded by humans.
…
You SHOULD complicate the given prompt using the following method: Please add one more constraints/requirements into #Given Prompt#

#Given Prompt#:
<Here is instruction.>
#Rewritten Prompt#:

Instruction Evolver – In-Breadth Evolution

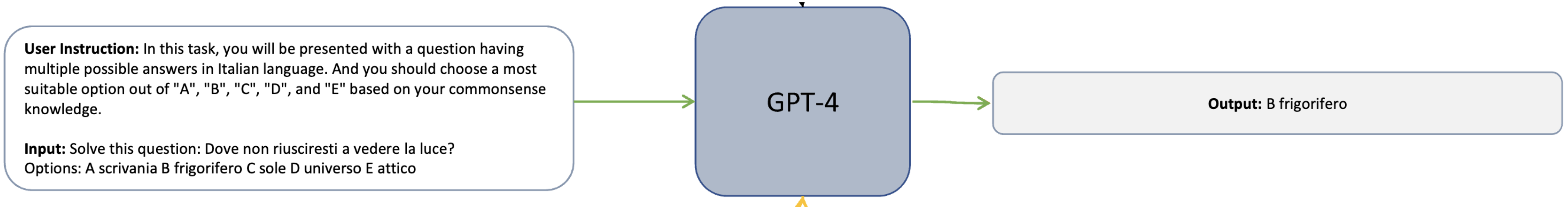
- Enhance
  - Topic Coverage
  - Skill Coverage

I want you act as a Prompt Creator. Your goal is to draw inspiration from the #Given Prompt# to create a brand new prompt. This new prompt should belong to the same domain as the #Given Prompt# but be even more rare. The LENGTH and difficulty level of the #Created Prompt# should be similar to that of the #Given Prompt#.
The #Created Prompt# must be reasonable and must be understood and responded by humans. '#Given Prompt#', '#Created Prompt#', 'given prompt' and 'created prompt' are not allowed to appear in #Created Prompt#.

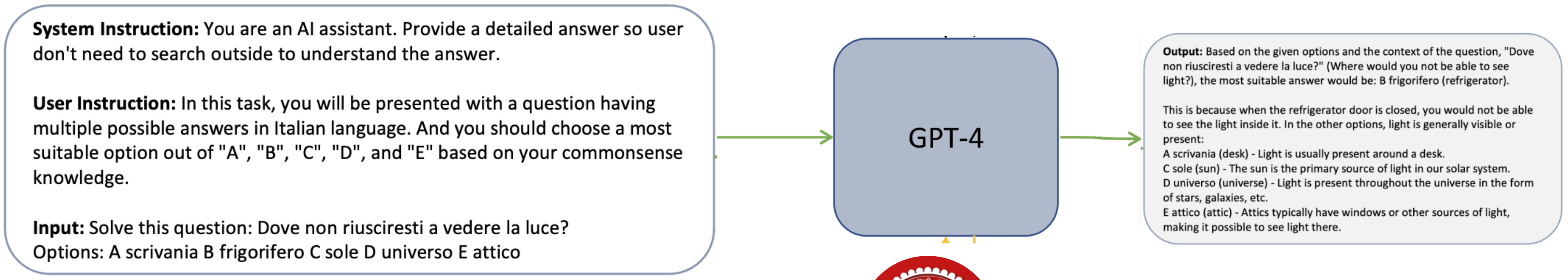
#Given Prompt#:
<Here is instruction.>
#Created Prompt#:

Orca

- How can we improve the information content in the response?



- Add a system instruction from a diverse instruction set including chain-of-thought, reasoning steps, explain like I'm five, being helpful and informative, etc.



Instruction Back-Translation

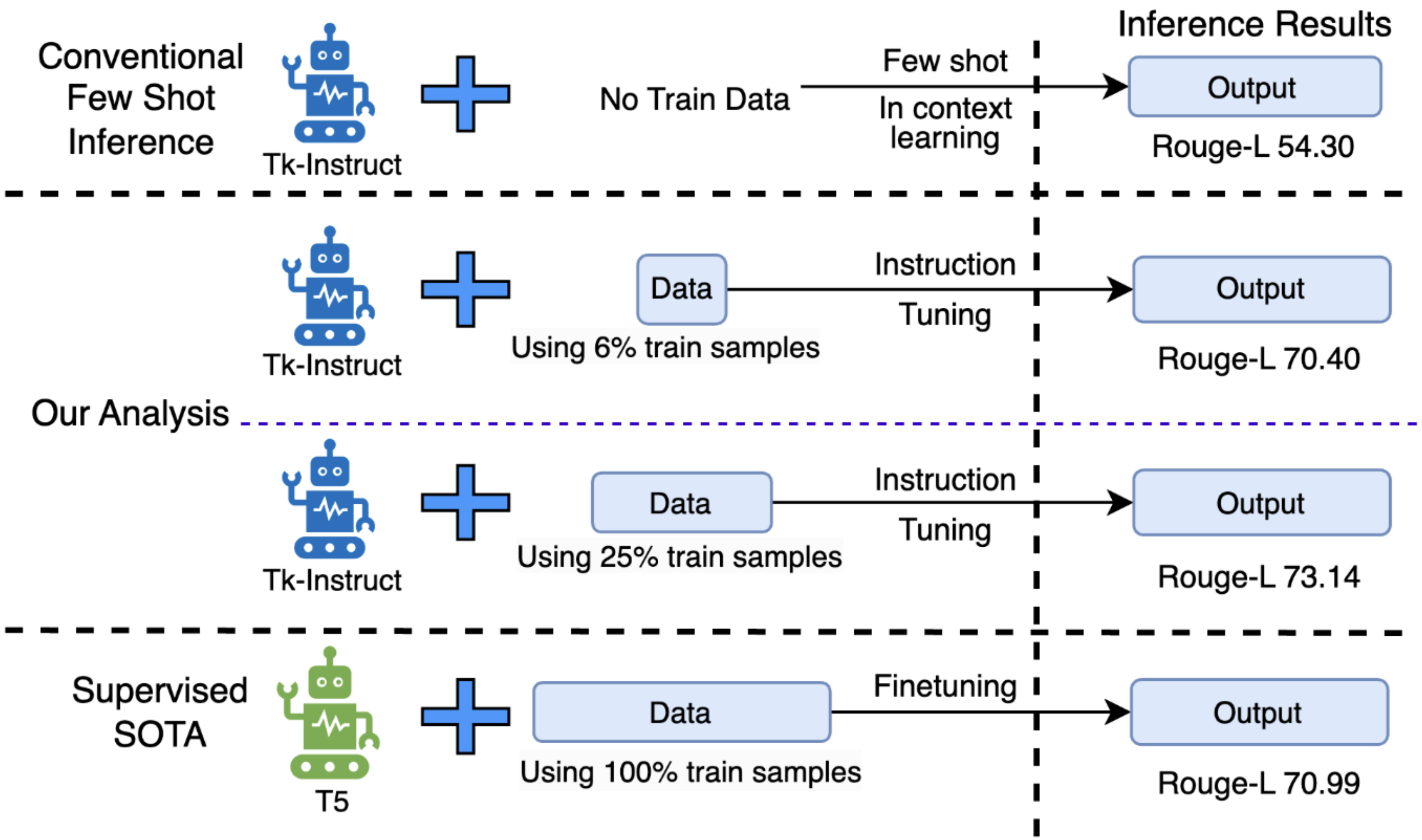
Gather Data:	Back-Translation Model Training:	Generate Raw Data:	Evaluation Model:	Filter & Fine-Tune:
• Unlabeled text from Clueweb (Overwijk et al., 2022). • 3,200 pieces of human-written (instruction, response) format data.	• Use LLaMA (Touvron et al., 2023b). • Train on seed data, taking response as input, generating instruction as output.	• Feed unlabeled texts into the trained back-translation model. • Produce raw (instruction, response) format data.	• Train another LLaMA-based model. • Assess the quality of (instruction, response) pairs generated in Step 3.	• Remove low-quality pairs. • Use remaining data for fine-tuning large language models (LLMs).

Content Credit: Instruction Tuning for Large Language Models: A Survey

Popular Instruction-Tuned Models on Known Datasets

- Flan-T5 (11B)
  - Fine-tuned T5-11B on Flan dataset
- Alpaca (7B)
  - Finetuned LLaMa-7B on synthetic dataset generated from text-davinci-003 generated using Self-Instruct
- WizardLM (7B)
  - Finetuned LLaMa-7B on on an instruction dataset generated from ChatGPT using Evol-Instruct .
- Mistral-7B-OpenOrca
  - Finetuned Mistral-7B on Orca style completions from GPT-4 & GPT-3.5

Instruction Tuned Models are Quick Learners



Superficial Alignment Hypothesis

- A model's knowledge and capabilities are learnt almost entirely during pretraining, while alignment teaches it which sub-distribution of formats should be used when interacting with users
- Corollary: A small number of examples should be sufficient for instruction-tuning.

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

Credit: LIMA: Less Is More for Alignment

LIMA (Less is More For Alignment)

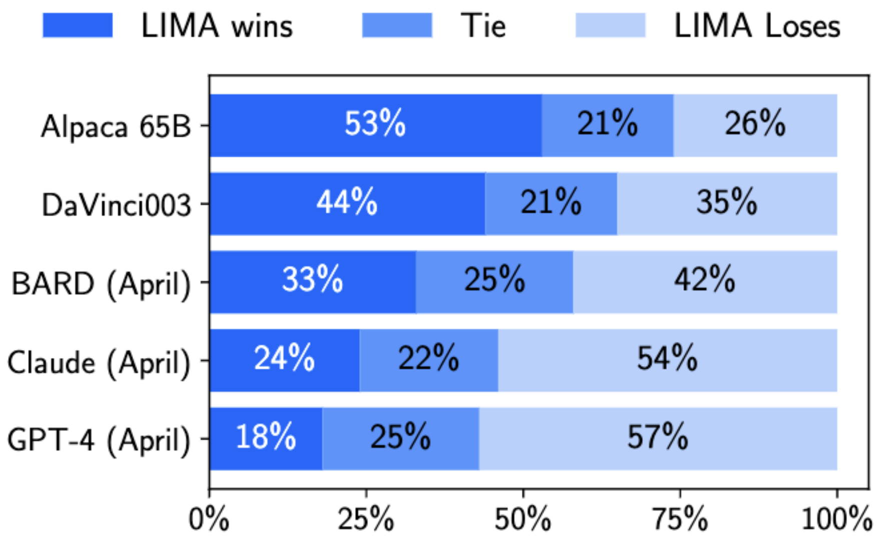


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

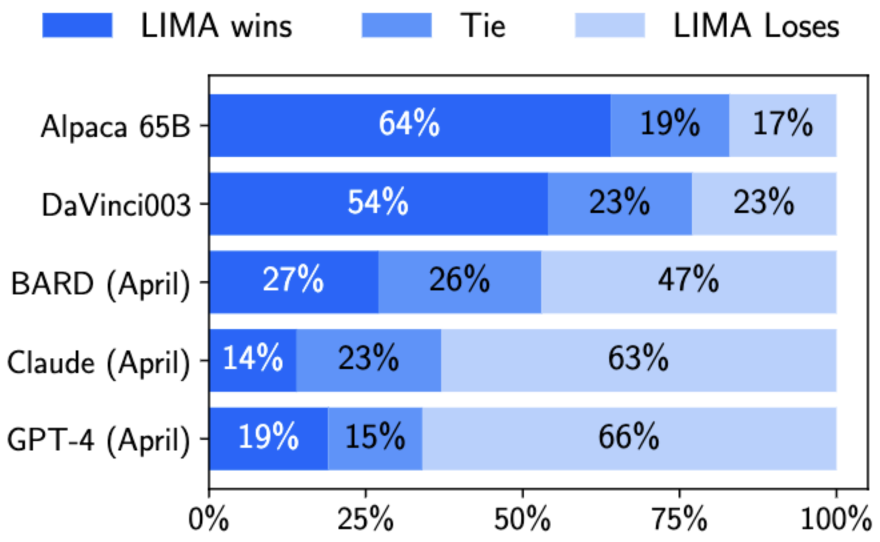


Figure 2: Preference evaluation using GPT-4 as the annotator, given the same instructions provided to humans.

Main Takeaways

Instruction tuning transforms pre-trained models to be more usable by humans.

Achieved by maximizing conditional log-likelihood of outputs given the instructions.

Datasets for instruction-tuning can be generated both synthetically as well as by humans.

Instruction-tuned models can quickly learn a task with limited data.