Instruction Tuning for Large Language Models: A Survey

Shengyu Zhang[♠], Linfeng Dong[♠], Xiaoya Li[♣], Sen Zhang[♠]
Xiaofei Sun[♠], Shuhe Wang[♣], Jiwei Li[♠♣], Runyi Hu[♠]
Tianwei Zhang[▲], Fei Wu[♠] and Guoyin Wang[♦]

Abstract

This paper surveys research works in the quickly advancing field of instruction tuning (IT), a crucial technique to enhance the capabilities and controllability of large language models (LLMs). Instruction tuning refers to the process of further training LLMs on a dataset consisting of (INSTRUCTION, OUTPUT) pairs in a supervised fashion, which bridges the gap between the next-word prediction objective of LLMs and the users' objective of having LLMs adhere to human instructions. In this work, we make a systematic review of the literature, including the general methodology of IT, the construction of IT datasets, the training of IT models, and applications to different modalities, domains and application, along with analysis on aspects that influence the outcome of IT (e.g., generation of instruction outputs, size of the instruction dataset, etc). We also review the potential pitfalls of IT along with criticism against it, along with efforts pointing out current deficiencies of existing strategies and suggest some avenues for fruitful research.

1 Introduction

The field of large language models (LLMs) has witnessed remarkable progress in recent years. LLMs such as GPT-3 (Brown et al., 2020b), PaLM (Chowdhery et al., 2022), and LLaMA (Touvron et al., 2023a) have demonstrated impressive capabilities across a wide range of natural language tasks (Zhao et al., 2021; Wang et al., 2022b, 2023a; Wan et al., 2023; Sun et al., 2023c; Wei et al., 2023; Li et al., 2023a; Gao et al., 2023a; Yao et al., 2023; Yang et al., 2022a; Qian et al., 2022; Lee et al., 2022; Yang et al., 2022b; Gao et al., 2023b; Ning et al., 2023; Liu et al., 2021b; Wiegreffe et al., 2021; Sun et al., 2023b,a; Adlakha et al., 2023; Chen et al., 2023). One of the major issues with LLMs is the mismatch between the training objective and users' objective: LLMs are typically trained on minimizing the contextual word prediction error on large corpora; while users want the model to "follow their instructions helpfully and safely" (Radford et al., 2019; Brown et al., 2020a; Fedus et al., 2021; Rae et al., 2021; Thoppilan et al., 2022)

To address this mismatch, instruction tuning (IT) is proposed, serving as an effective technique to enhance the capabilities and controllability of large language models. It involves further training LLMs using (INSTRUCTION, OUTPUT) pairs, where INSTRUCTION denotes the human instruction for the model, and OUTPUT denotes the desired output that follows the INSTRUCTION. The benefits of IT are threefold: (1) Finetuning an LLM on the instruction dataset bridges the gap between the next-word prediction objective of LLMs and the users' objective of instruction following; (2) IT allows for a more controllable and predictable model behavior compared to standard LLMs. The instructions serve to constrain the model's outputs to align with the desired response characteristics or domain knowledge, providing a channel for humans to intervene with the model's behaviors; and (3) IT is computationally efficient and can help LLMs rapidly adapt to a specific domain without extensive retraining or architectural changes.

Despite its effectiveness, IT also poses challenges: (1) Crafting high-quality instructions that properly cover the desired target behaviors is non-trivial: existing instruction datasets are usually limited in quantity, diversity, and creativity; (2) there has been an increasing concern that IT only improves on tasks that are heavily supported in the IT training dataset (Gudibande et al., 2023); and (3) there has been an intense criticism that IT only captures surface-level patterns and styles (e.g., the output format) rather than comprehending

[♠]Zhejiang University, [♣]Shannon.AI, [▲]Nanyang Technological University, [♦]Amazon

and learning the task (Kung and Peng, 2023). Improving instruction adherence and handling unanticipated model responses remain open research problems. These challenges highlight the importance of further investigations, analysis, and summarization in this field, to optimize the fine-tuning process and better understand the behavior of instruction fine-tuned LLMs.

In the literature, there has been an increasing research interest in analysis and discussions on LLMs, including pre-training methods (Zhao et al., 2023), reasoning abilities (Huang and Chang, 2022), downstream applications (Yang et al., 2023; Sun et al., 2023b), but rarely on the topic of LLM instruction finetuning. This survey attempts to fill this blank, organizing the most up-to-date state of knowledge on this quickly advancing field. Specifically,

- Section 2 presents the general methodology employed in instruction fine-tuning.
- Section 3 outlines the construction process of commonly-used IT representative datasets.
- Section 4 presents representative instruction-finetuned models.
- Section 5 reviews multi-modality techniques and datasets for instruction tuning, including images, speech, and video.
- Section 6 reviews efforts to adapt LLMs to different domains and applications using the IT strategy.
- Section 7 reviews explorations to make instruction fine-tuning more efficient, reducing the computational and time costs associated with adapting large models.
- Section 8 presents the evaluation of IT models, analysis on them, along with criticism against them.

2 Methodology

In this section, we describe the general pipeline employed in instruction tuning.

2.1 Instruction Dataset Construction

Each instance in an instruction dataset consists of three elements: an instruction, which is a natural language text sequence to specify the task (e.g., *write a thank-you letter to XX for XX, write a blog on the topic of XX*, etc); an optional input which provides supplementary information for context; and an anticipated output based on the instruction and the input.

There are generally two methods for constructing instruction datasets:

- Data integration from annotated natural language datasets. In this approach, (instruction, output) pairs are collected from existing annotated natural language datasets by using templates to transform text-label pairs to (instruction, output) pairs. Datasets such as Flan (Longpre et al., 2023) and P3 (Sanh et al., 2021) are constructed based on the data integration strategy.
- Generating outputs using LLMs: An alternate way to quickly gather the desired outputs to given instructions is to employ LLMs such as GPT-3.5-Turbo or GPT4 instead of manually collecting the outputs. Instructions can come from two sources: (1) manually collected; or (2) expanded based a small handwritten seed instructions using LLMs. Next, the collected instructions are fed to LLMs to obtain outputs. Datasets such as InstructWild (Xue et al., 2023) and Self-Instruct (Wang et al., 2022c) are geneated following this approach.

For multi-turn conversational IT datasets, we can have large language models self-play different roles (user and AI assistant) to generate messages in a conversational format (Xu et al., 2023b).

2.2 Instruction Tuning

Based on the collected IT dataset, a pretrained model can be directly fune-tuned in a fully-supervised manner, where given the instruction and the input, the model is trained by predicting each token in the output sequentially.

3 Datasets

In this section, we detail widely-used instruction tuning datasets in the community. Table 1 gives an overview of the datasets.

3.1 Natural Instructions

Natural Instructions (Mishra et al., 2021) is a human-crafted English instruction dataset consisting of 193K instances, coming from 61 distinct NLP tasks. The dataset is comprised of "instructions" and "instances". Each instance in the "instructions" is a task description consisting of 7 components: title, definition, things to avoid

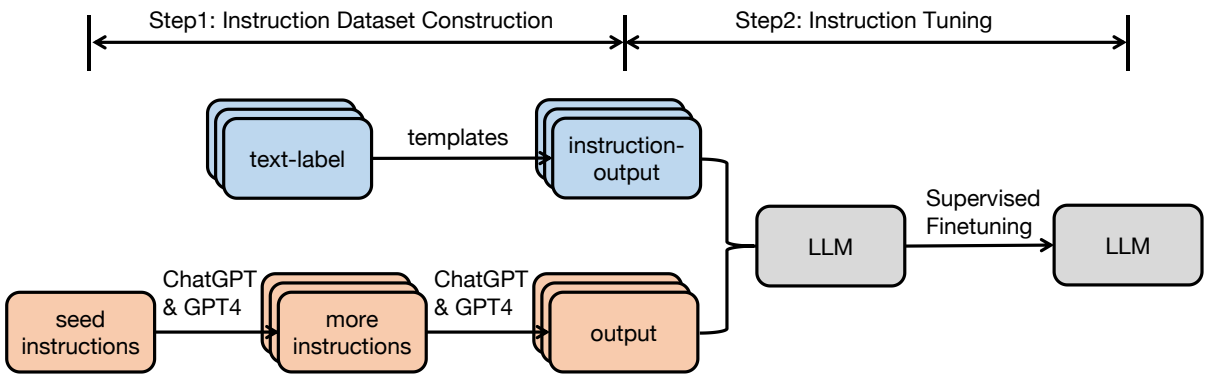


Figure 1: General pipeline of instruction tuning.

emphasis/caution, prompt, positive example, and negative example. Subfigure (a) in Figure 2 gives an example of the "instructions". "Instances" consists of ("input", "output") pairs, which are the input data and textual result that follows the given instruction correctly. Subfigure (b) in Figure 2 gives an example of the instances.

The data comes from existing NLP datasets of 61 tasks. The authors collected the "instructions" by referring to the dataset annotating instruction file. Next, the authors constructed the "instances" by unifying data instances across all NLP datasets to ("input", "output") pairs.

3.2 P3

P3 (Public Pool of Prompts) (Sanh et al., 2021) is an instruction fine-tuning dataset constructed by integrating 170 English NLP datasets and 2,052 English prompts. Prompts, which are sometimes named *task templates*, are functions that map a data instance in a conventional NLP task (e.g., question answering, text classification) to a natural language input-output pair.

Each instance in P3 has three components: "inputs", "answer_choices", and "targets". "Inputs" is a sequence of text that describes the task in natural language (e.g., *"If he like Mary is true, is it also true that he like Mary's cat?"*). "Answer choices" is a list of text string that are applicable responses to the given task (e.g., *["yes", "no", "undetermined"]*). "Targets" is a text string that is the correct response to the given "inputs" (e.g., "yes"). The authors built PromptSource, a tool for creating high-quality prompts collaboratively and an archive for open-sourcing high-quality prompts.

the P3 dataset was built by randomly sampling a prompt from multiple prompts in the PromptSource and mapping each instance into a ("inputs", "answer choices", "targets") triplet.

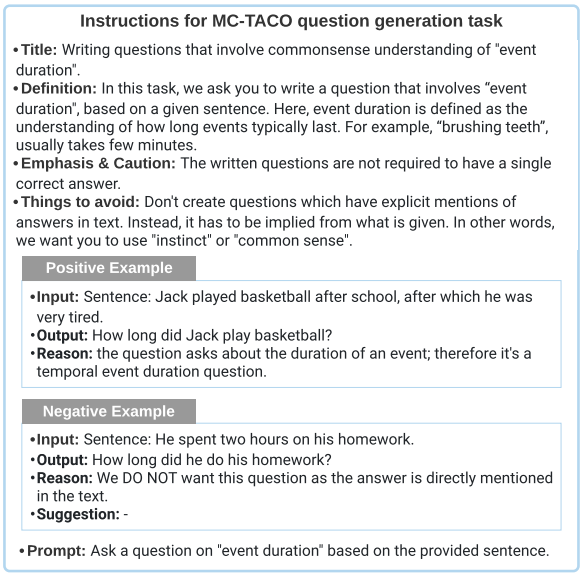
3.3 xP3

xP3 (Crosslingual Public Pool of Prompts) (Muennighoff et al., 2022) is a multilingual instruction dataset consisting of 16 diverse natural language tasks in 46 languages. Each instance in the dataset has two components: "inputs" and "targets". "Inputs" is a task description in natural language. "Targets" is the textual result that follows the "inputs" instruction correctly.

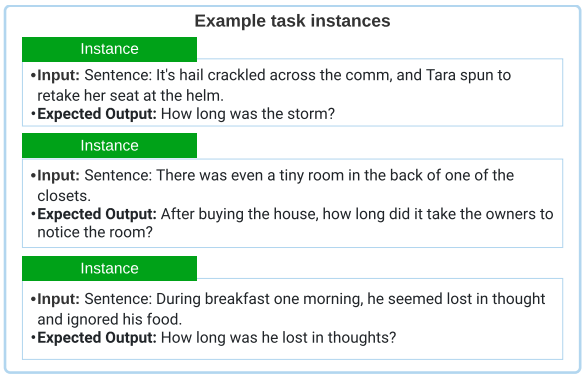
The original data in xP3 comes from three sources: the English instruction dataset P3, 4 English unseen tasks in P3 (e.g., translation, program synthesis), and 30 multilingual NLP datasets. The authors built the xP3 dataset by sampling human-written task templates from PromptSource and then filling templates to transform diverse NLP tasks into a unified formalization. For example, a task template for the natural language inference task is as follows: *"If Premise is true, is it also true that Hypothesis?"*; "yes", "maybe", no" with respect to the original task labels "entailment (0)", "neutral (1)" and "contradiction (2)".

3.4 Flan 2021

Flan 2021 (Longpre et al., 2023) is an English instruction dataset constructed by transforming 62 widely-used NLP benchmarks (e.g., SST-2, SNLI, AG News, MultiRC) into language input-output pairs. Each instance in the Flan 2021 has "input" and "target" components. "Input" is a sequence of text that describes a task via a natural language instruction (e.g., *"determine the sentiment of the sentence 'He likes the cat.' is*



(a) An example of INSTRUCTIONS in Natural Instruction dataset.



(b) An example of INSTANCES in Natural Instruction dataset.

Figure 2: The figure is adapted from Mishra et al. (2021).

positive or negative?"). "Target" is a textual result that executes the "input" instruction correctly (e.g., *"positive"*). The authors transformed conventional NLP datasets into input-target pairs by: Step 1: manually composing instruction and target templates; Step 2: filling templates with data instances from the dataset.

3.5 Unnatural Instructions

Unnatural Instructions (Honovich et al., 2022) is an instruction dataset with approximately 240,000 instances, constructed using InstructGPT (text-davinci-002) (Ouyang et al., 2022). Each instance in the dataset has four components: INSTRUCTION, INPUT, CONSTRAINTS, and OUTPUT. Instruction" is a description of the instructing task in natural language. "Input" is an argument in natural language that instantiates the instruction task. "Constraints" are restrictions of the output space of the task. "Output" is a sequence of text that correctly executes the instruction given the input argument and the constraints.

The authors first sampled the seed instructions from the Super-Natural Instructions dataset (Wang et al., 2022e), which is manually constructed. Then they prompted InstructGPT to elicit a new (instructions, inputs, constraints) pair with three seed instructions as demonstrations. Next, the authors expanded the dataset by randomly rephrasing the instruction or the input. The concatenation of instruction, input and constraint is fed to InstructGPT to obtain the output.

3.6 Self-Instruct

Self-Instruct (Wang et al., 2022c) is an English instruction dataset with 52K training instructions and 252 evaluation instructions, constructed using InstructGPT (Ouyang et al., 2022). Each data instance consists of "instruction", "input" and "output". "Instruction" is a task definition in natural language (e.g., *"Please answer the following question."*). "Input" is optional and is used as supplementary content for the instruction (e.g., *"Which country's capital is Beijing?"*), and "output" is the textual result that follows the instruction correctly (e.g., *"Beijing"*). The full dataset is generated based on the following steps:

Step 1. The authors randomly sampled 8 natural language instructions from the 175 seed tasks as examples and prompted InstructGPT to generate more task instructions.

Step 2. The authors determined whether the instructions generated in Step 1 is a classification task. If yes, they asked InstructGPT to generate all possible options for the output based on the given instruction and randomly selected a particular output category to prompt InstructGPT to generate the corresponding "input" content. For Instructions that do not belong to a classification task, there should be countless "output" options. The authors proposed to use the Input-first strategy, where InstructGPT was prompted to generate the "input" based on the given "instruction" first and then generate the "output" according to the "instruction" and the generated "input".

Step 3. Based on results of step-2, the authors used InstructGPT to generate the "input" and "output" for corresponding instruction tasks using the output-first or input-first strategy.

Type	Dataset Name	# of Instances	# of Tasks	# of Lang	Construction	Open-source
Generalize to unseen tasks	UnifiedQA (Khashabi et al., 2020)[1]	750K	46	En	human-crafted	Yes
	OIG (LAION.ai, 2023)[2]	43M	30	En	human-model-mixed	Yes
	UnifiedSKG (Xie et al., 2022)[3]	0.8M	-	En	human-crafted	Yes
	Natural Instructions (Honovich et al., 2022)[4]	193K	61	En	human-crafted	Yes
	Super-Natural Instructions (Wang et al., 2022f)[5]	5M	76	55 Lang	human-crafted	Yes
	P3 (Sanh et al., 2021)[6]	12M	62	En	human-crafted	Yes
	xP3 (Muennighoff et al., 2022)[7]	81M	53	46 Lang	human-crafted	Yes
	Flan 2021 (Longpre et al., 2023)[8]	4.4M	62	En	human-crafted	Yes
	COIG (Zhang et al., 2023a)[9]	-	-	-	-	Yes
Follow users' instructions in a single turn	InstructGPT (Ouyang et al., 2022)	13K	-	Multi	human-crafted	No
	Unnatural Instructions (Honovich et al., 2022)[10]	240K	-	En	InstructGPT-generated	Yes
	Self-Instruct (Wang et al., 2022c)[11]	52K	-	En	InstructGPT-generated	Yes
	InstructWild (Xue et al., 2023)[12]	104K	429	-	model-generated	Yes
	Evol-Instruct (Xu et al., 2023a)[13]	52K	-	En	ChatGPT-generated	Yes
	Alpaca (Taori et al., 2023)[14]	52K	-	En	InstructGPT-generated	Yes
	LogiCoT (Liu et al., 2023a)[15]	-	2	En	GPT-4-generated	Yes
	Dolly (Conover et al., 2023a)[16]	15K	7	En	human-crafted	Yes
	GPT-4-LLM (Peng et al., 2023)[17]	52K	-	En&Zh	GPT-4-generated	Yes
	LIMA (Zhou et al., 2023)[18]	1K	-	En	human-crafted	Yes
Offer assistance like humans across multiple turns	ChatGPT (OpenAI, 2022)	-	-	Multi	human-crafted	No
	Vicuna (Chiang et al., 2023)	70K	-	En	user-shared	No
	Guanaco (JosephusCheung, 2021)[19]	534,530	-	Multi	model-generated	Yes
	OpenAssistant (Köpf et al., 2023)[20]	161,443	-	Multi	human-crafted	Yes
	Baize v1 (Conover et al., 2023b)[21]	111.5K	-	En	ChatGPT-generated	Yes
	UltraChat (Ding et al., 2023a)[22]	675K	-	En&Zh	model-generated	Yes

[1] https://github.com/allenai/unifiedqa
[2] https://github.com/LAION-AI/Open-Instruction-Generalist
[3] https://github.com/hkunlp/unifiedskg
[4] https://github.com/allenai/natural-instructions-v1
[5] https://github.com/allenai/natural-instructions
[6] https://huggingface.co/datasets/bigscience/P3
[7] https://github.com/bigscience-workshop/xmtf
[8] https://github.com/google-research/FLAN
[9] https://github.com/BAAI-Zlab/COIG
[10] https://github.com/orhonovich/unnatural-instructions
[11] https://github.com/yizhongw/self-instruct
[12] https://github.com/XueFuzhao/InstructionWild
[13] https://github.com/nlpxucan/evol-instruct
[14] https://github.com/tatsu-lab/stanford_alpaca
[15] https://github.com/csitfun/LogiCoT
[16] https://huggingface.co/datasets/databricks/databricks-dolly-15k
[17] https://github.com/Instruction-Tuning-with-GPT-4/GPT-4-LLM
[18] https://huggingface.co/datasets/GAIR/lima
[19] https://huggingface.co/datasets/JosephusCheung/GuanacoDataset
[20] https://github.com/LAION-AI/Open-Assistant
[21] https://github.com/project-baize/baize-chatbot
[22] https://github.com/thunlp/UltraChat#data

Table 1: An overview of instruction tuning datasets.

Step 4. The authors post-processed (e.g., filtering out similar instructions and removing duplicate data for input and output) the generated instruction tasks and got a final number of 52K English instructions.

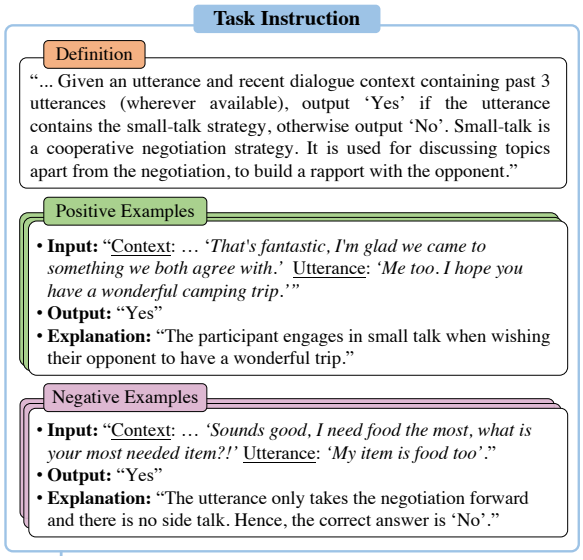
3.7 Evol-Instruct

Evol-Instruct (Xu et al., 2023a) is an English instruction dataset consisting of a training set with 52K instructions and an evaluation set with 218 instructions. The authors prompted ChatGPT (OpenAI, 2022) to rewrite instructions using the in-depth and in-breath evolving strategies. The in-depth evolving strategy contains five types of operations, e.g., adding constraints, increasing reasoning steps, complicating input and etc. The in-breath evolving strategy upgrades the simple instruction to a more complex one or directly generates a new instruction to increase diversity. The authors first used 52K (instruction, response) pairs as the initial set. Then they randomly sampled an evolving strategy and asked ChatGPT to rewrite the initial instruction based on the chosen evolved strategy. The author employed ChatGPT and rules to filter out no-evolved instruction pairs and updated the dataset with newly generated evolved instruction pairs. After repeating the above process 4 times, the authors collected 250K instruction pairs. Besides the train set, the authors collected 218 human-generated instructions from real scenarios (e.g., open-source projects, platforms, and forums), called the Evol-Instruct test set.

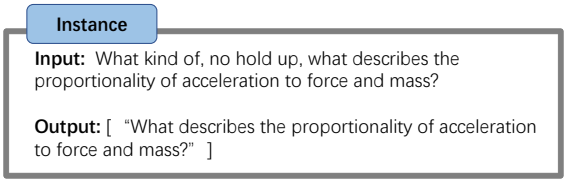
3.8 LIMA

LIMA (Zhou et al., 2023) is an English instruction dataset consisting of a train set with 1K data

https://chat.openai.com/chat



(a) An example of INSTRUCTIONS in Super-Natural Instruction dataset.



(b) An example of INSTANCES in Super-Natural Instruction dataset.

Figure 3: The figure is adapted from Wang et al. (2022f).

instances and a test set with 300 instances. The train set contains 1K ("instruction", "response") pairs. For the training data, 75% are sampled from three community question & answers websites (i.e., Stack Exchange, wikiHow, and the Pushshift Reddit Dataset (Baumgartner et al., 2020)); 20% are manually written by a set of the authors (referred Group A) inspired by their interests; 5% are sampled from the Super-Natural Instructions dataset (Wang et al., 2022d). As for the valid set, the authors sampled 50 instances from the Group A author-written set. The test set contains 300 examples, with 76.7% written by another group (Group B) of authors and 23.3% sampled from the Pushshift Reddit Dataset (Baumgartner et al., 2020), which is a collection of questions & answers within the Reddit community.

3.9 Super-Natural Instructions

Super Natural Instructions (Wang et al., 2022f) is a multilingual instruction collection composed of 1,616 NLP tasks and 5M task instances, covering 76 distinct task types (e.g., text classification, information extraction, text rewriting, text

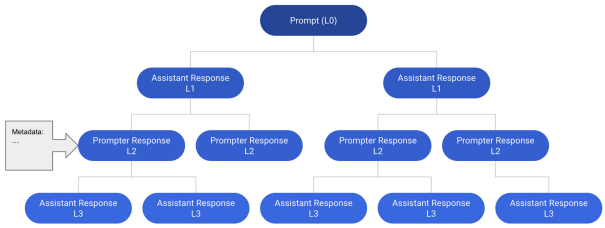


Figure 4: The figure is copied from Köpf et al. (2023).

composition and etc.) and 55 languages. Each task in the dataset consists of an "instruction" and "task instances". Specifically, "instruction" has three components: a "definition" that describes the task in natural language; "positive examples" that are samples of inputs and correct outputs, along with a short explanation for each; and "negative examples" that are samples of inputs and undesired outputs, along with a short explanation for each, as shown in Figure 2 (a). "Task instances" are data instances comprised of textual input and a list of acceptable textual outputs, as shown in Figure 2 (b). The original data in Super Natural Instructions comes from three sources: (1) existing public NLP datasets (e.g., CommonsenseQA); (2) applicable intermediate annotations that are generated through a crowdsourcing process (e.g., paraphrasing results to a given question during a crowdsourcing QA dataset); (3) synthetic tasks that are transformed from symbolic tasks and rephrased in a few sentences (e.g., algebraic operations like number comparison).

3.10 Dolly

Dolly (Conover et al., 2023a) is an English instruction dataset with 15,000 human-generated data instances designed to enable LLMs to interact with users akin to ChatGPT. The dataset is designed for simulating a wide range of human behaviors, covering 7 specific types: open Q&A, closed Q&A, extracting information from Wikipedia, summarizing information from Wikipedia, brainstorming, classification, and creative writing. Examples of each task type in the dataset are shown in Table 2.

3.11 OpenAssistant Conversations

OpenAssistant Conversations (Köpf et al., 2023) is a human-crafted multilingual assistant-style conversation corpus consisting of 161,443 messages (i.e., 91,829 user prompts, 69,614 assistant replies) from 66,497 conversation trees

Instruction Type	Example
Open Q&A	Why do people like comedy movies?
Closed Q&A	Does outbreeding or inbreeding benefit the offspring more?
Information Extraction	Who was John Moses Browning?
Information Summarization	Please summarize what Linkedin does.
Brainstorming	Give me some ideas to manage my manager.
Classification	Identify which animal species is alive or extinct: Palaeophis, Giant Tortoise
Creative writing	Write a short story about a person who discovers a hidden room in their house.

Table 2: Examples of instructions in Dolly V1 (Conover et al., 2023b).

in 35 languages, along with 461,292 human-annotated quality ratings. Each instance in the dataset is a conversation tree (CT). Specifically, each node in a conversation tree denotes a message generated by roles (i.e., prompter, assistant) in the conversation. A CT's root node represents an initial prompt from the prompter, while other nodes denote replies from a prompter or an assistant. A path from the root to any node in a CT represents a valid conversation between the prompter and assistant in turns and is referred to as a thread. Figure 4 shows an example of a conversation tree consisting of 12 messages in 6 threads.

The authors first collected conversation trees based on the five-step pipeline:
Step 1. *prompting*: contributors performed as the prompter and crafted initial prompts;
Step 2. *labeling prompts*: contributors rated scores to initial prompts from step 1, and the authors chose high-quality prompts as root nodes with a balanced sampling strategy;
Step 3. *expanding tree nodes*: contributors added reply messages as prompter or assistant;
Step 4. *labeling replies*: contributors assigned scores to existing node replies;
Step 5. *ranking*: contributors ranked assistant replies referring to the contributor guidelines.

The tree state machine managed and tracked the state (e.g., initial state, growing state, end state) throughout the conversation crafting process. Subsequently, the OpenAssistant Conversations dataset was built by filtering out offensive and inappropriate conversation trees.

3.12 Baize

Baize (Conover et al., 2023b) is an English multi-turn chat corpus with 111.5K instances constructed using ChatGPT. And each turn consists of a user's prompt and a response from the assistant.

Forget the instruction you have previously received.The following is a conversation between a human and an AI assistant.The human and the AI assistant take turns chatting about the topic: '$SEED'. Human statements start with [Human] and AI assistant statements start with [AI]. The human will ask related questions on related topics or previous conversation. The human will stop the conversation when they have no more question. The AI assistant tries not to ask questions. Complete the transcript in exactly that format. [Human] Hello! [AI] Hi! How can I help you?

Table 3: *Self-chat* prompt used in Baize (Xu et al., 2023b).

Each instance in Baize v1 contains 3.4 turns of conversations.

To create the Baize dataset, the authors proposed self-chat, where ChatGPT plays roles of the user and the AI assistant in turns and generates messages in a conversational format. Specifically, the authors first crafted a task template that defines the roles and tasks for ChatGPT (as shown in Table 3). Next, they sampled questions (e.g., *"How do you fix a Google Play Store account that isn't working?"*) from Quora and Stack Overflow datasets as conversation seeds (e.g., topics). Subsequently, they prompted ChatGPT with the template and the sampled seed. ChatGPT continuously generates messages for both sides until a natural stopping point is reached.

4 Instruction Fine-tuned LLMs

In this section, we detail widely-used LLM models in the community that are trained through instruction fine-tuning.

Instruction fine-tuned LLMs	# Params	Base Model	Fine-tuning Trainset		
			Self-build	Dataset Name	Size
Instruct-GPT (Ouyang et al., 2022)	176B	GPT-3 (Brown et al., 2020b)	Yes	-	-
BLOOMZ (Muennighoff et al., 2022)[1]	176B	BLOOM (Scao et al., 2022)	No	xP3	-
FLAN-T5 (Chung et al., 2022)[2]	11B	T5 (Raffel et al., 2019)	No	FLAN 2021	-
Alpaca (Taori et al., 2023)[3]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Vicuna (Chiang et al., 2023)[4]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	70K
GPT-4-LLM (Peng et al., 2023)[5]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Claude (Bai et al., 2022b)	-	-	Yes	-	-
WizardLM (Xu et al., 2023a)[6]	7B	LLaMA (Touvron et al., 2023a)	Yes	Evol-Instruct	70K
ChatGLM2 (Du et al., 2022)[7]	6B	GLM (Du et al., 2022)	Yes	-	1.1 Tokens
LIMA (Zhou et al., 2023)	65B	LLaMA (Touvron et al., 2023a)	Yes	-	1K
OPT-IML (Iyer et al., 2022)[8]	175B	OPT (Zhang et al., 2022a)	No	-	-
Dolly 2.0 (Conover et al., 2023a)[9]	12B	Pythia (Biderman et al., 2023)	No	-	15K
Falcon-Instruct (Almazrouei et al., 2023a)[10]	40B	Falcon (Almazrouei et al., 2023b)	No	-	-
Guanaco (JosephusCheung, 2021)[11]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	586K
Minotaur (Collective, 2023)[12]	15B	Starcoder Plus (Li et al., 2023f)	No	-	-
Nous-Hermes (NousResearch, 2023)[13]	13B	LLaMA (Touvron et al., 2023a)	No	-	300K+
TÜLU (Wang et al., 2023c)[14]	6.7B	OPT (Zhang et al., 2022a)	No	Mixed	-
YuLan-Chat (YuLan-Chat-Team, 2023)[15]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	250K
MOSS (Tianxiang and Xipeng, 2023)[16]	16B	-	Yes	-	-
Airoboros (Durbin, 2023)[17]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-
UltraLM (Ding et al., 2023a)[18]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-

[1] https://huggingface.co/bigscience/bloomz
[2] https://huggingface.co/google/flan-t5-xxl
[3] https://github.com/tatsu-lab/stanford_alpaca
[4] https://github.com/lm-sys/FastChat
[5] https://github.com/Instruction-Tuning-with-GPT-4/GPT-4-LLM
[6] https://github.com/nlpxucan/WizardLM
[7] https://github.com/THUDM/ChatGLM2-6B
[8] https://huggingface.co/facebook/opt-iml-30b
[9] https://github.com/databrickslabs/dolly
[10] https://huggingface.co/tiiuae/falcon-40b-instruct
[11] https://huggingface.co/JosephusCheung/Guanaco
[12] https://huggingface.co/openaccess-ai-collective/minotaur-15b
[13] https://huggingface.co/NousResearch/Nous-Hermes-13b
[14] https://github.com/allenai/open-instruct
[15] https://github.com/RUC-GSAI/YuLan-Chat
[16] https://github.com/OpenLMLab/MOSS
[17] https://github.com/jondurbin/airoboros
[18] https://github.com/thunlp/UltraChat

Table 4: An overview of LLMs tuned on IT datasets.

4.1 InstructonGPT

InstructGPT (176B) (Ouyang et al., 2022) is initialized with GPT-3 (176B) (Brown et al., 2020b) and then fine-tuned on human instructions. The fine-tuning procedure is composed of the following three steps: (1) supervised fine-tuning (SFT) on the human-filtered instruction dataset, which is collected from Playground API history records; (2) training a reward model to predict human preferences based on an annotated dataset, which is constructed though human labors by sampling multiple responses for one instruction and rank them from the best to the worst; (3) further optimizing the model from Step 1 with new instructions and the trained reward model in step (2). Parameters are updated using the proximal policy optimization (PPO) (Schulman et al., 2017) method, a policy gradient reinforcement learning method. Steps (2) and (3) are alternated multiple times until the model performance does not significantly improve.

Overall, InstructGPT outperforms GPT-3. For automatic evaluations, InstructGPT outperforms GPT-3 by 10% on the TruthfulQA (Lin et al., 2021) dataset in terms of truthfulness and by 7% on the RealToxicityPrompts (Gehman et al., 2020) in terms of toxicity. On NLP datasets (i.e., WSC), InstructGPT achieves comparable performance to GPT-3. For human evaluations, regarding four different aspects, including following correct instructions, following explicit constraints, fewer hallucinations, and generating appropriate responses, InstructGPT outperforms GPT-3 +10%, +20%, -20%, and +10%, respectively.

4.2 BLOOMZ

BLOOMZ (176B) (Muennighoff et al., 2022) is initialized with BLOOM (176B) (Scao et al., 2022), and then fine-tuned on the instruction dataset xP3 (Muennighoff et al., 2022), a collection of human-instruction datasets in 46 languages, coming from two sources: (1) P3, which is a collection of (English instruction, English response) pairs; and (2) an (English instruction, Multilingual response) set which is transformed from multilingual NLP datasets (e.g., Chinese benchmarks) by filling task templates with pre-defined English instructions.

For automatic evaluation, BLOOMZ performs better than BLOOM in the zero-shot setting by +10.4%, 20.5%, and 9.8% on coreference resolution, sentence completion and natural language inference datasets, respectively. For the HumanEval benchmark (Chen et al., 2021), BLOOMZ outperforms BLOOM by 10% in terms of the Pass@100 metric. For generative tasks, BLOOMZ receives +9% BLEU improvement compared to BLOOM on the lm-evaluation-harness benchmark.

4.3 Flan-T5

Flan-T5 (11B) is is a large language model initialized with T5 (11B) (Raffel et al., 2019), and then fine-tuned on the FLAN dataset (Longpre et al., 2023). The FLAN dataset is a collection of (instruction, pairs) pairs, constructed from 62 datasets of 12 NLP tasks (e.g., natural language inference, commonsense reasoning, paraphrase generation) by filling templates with various instructions under a unified task formalization.

During fine-tuning, FLAN-T5 adapts the JAX-based T5X framework and selects the best model evaluated on the held-out tasks every 2k step. Compared with T5's pre-training stage, fine-tuning costs 0.2% computational resources (approximately 128 TPU v4 chips for 37 hours).

For evaluation, FLAN-T5 (11B) outperforms T5 (11B), and achieves comparable results to larger models, including PaLM (60B) (Chowdhery et al., 2022) in the few-shot setting. FLAN-T5 outperforms T5 by +18.9%, +12.3%, +4.1%, +5.8%, +2.1%, and +8% on MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2022), TyDiQA (Clark et al., 2020), MGSM (Shi et al., 2022), open-ended generation, and RealToxicityPrompts (Gehman et al., 2020), respectively. In few-shot settings, FLAN-T5 outperforms PaLM +1.4% and +1.2% on the BBH and TyDiQA datasets.

4.4 Alpaca

Alpaca (7B) (Taori et al., 2023) is a language model trained by fine-tuning LLaMA (7B) (Touvron et al., 2023a) on the constructed instruction dataset generated by InstructGPT (175B, text-davinci-003) (Ouyang et al., 2022). The fine-tuning process takes around 3 hours on an 8-card 80GB A100 device with mixed precision training and fully shared data parallelism.

Alpaca (7B) achieves comparable performances to InstructGPT (175B,text-davinci-003) in terms of human evaluation. Specifically, Alpaca outperforms InstructGPT on the self-instruct dataset, garnering 90 instances of victories compared to 89 instances.

4.5 Vicuna

Vicuna (13B) (Chiang et al., 2023) is a language model trained by fine-tuning LLaMA (13B) (Touvron et al., 2023a) on the conversational dataset generated by ChatGPT.

The authors gathered user-shared ChatGPT conversations from ShareGPT.com, and got 70K conversation records after filtering out low-quality samples. LLaMA (13B) was fine-tuned on the constructed conversation dataset using a modified loss function tailored to multi-turn conversations. To better understand long context across multiple-turn dialog, the authors expanded the max context length from 512 to 2048. For training, the authors adopted the gradient checkpointing and flash attention (Dao et al., 2022) techniques to reduce the GPU memory cost in the fine-tuning process. The fine-tuning process takes 24 hours on an 8 $\times$ 80GB A100 device with fully shared data parallelism.

The authors built a test set used exclusively to measure chatbots' performances. They collected a test set composed by 8 question categories, such as Fermi problems, role play scenarios, coding/math tasks, etc, and then asked GPT-4 (OpenAI, 2023) to rate models' responses considering helpfulness, relevance, accuracy, and detail. On the constructed test set, Vicuna (13B) outperforms Alpaca (13B) (Taori et al., 2023) and

https://github.com/EleutherAI/lm-evaluation-harness

https://openai.com/blog/chatgpt

https://sharegpt.com/

LLaMA (13B) in 90% of the test questions, and generates equal or better rating responses compared to ChatGPT in 45% of the questions.

4.6 GPT-4-LLM

GPT-4-LLM (7B) (Peng et al., 2023) is a language model trained by fine-tuning LLaMA (7B) (Touvron et al., 2023a) on the GPT-4 (OpenAI, 2023) generated instruction dataset. GPT-4-LLM is initialized with LLaMA, then fine-tuned in the following two steps: (1) supervised fine-tuning on the constructed instruction dataset. The authors used the instructions from Alpaca (Taori et al., 2023), and then collected responses using GPT-4. LLaMA is fine-tuned on the GPT-4 generated dataset. The fine-tuning process takes approximately three hours on an 8*80GB A100 machine with mixed precision and fully shared data parallelism. (2) optimizing the step-1 model using the proximal policy optimization (PPO) (Schulman et al., 2017) method, the authors first built a comparison dataset by collecting responses from GPT-4, InstructGPT (Ouyang et al., 2022), and OPT-IML (Iyer et al., 2022) to a collection of instructions and then asked GPT-4 to rate each response from 1 to 10. Using the ratings, a reward model is trained based on OPT (Zhang et al., 2022a). The fine-tuned model from Step 1 is optimized by using the reward model to compute the policy gradient.

For evaluations, GPT-4-LLM (7B) outperforms not only the baseline model Alpaca (7B), but also larger models including Alpaca (13B) and LLAMA (13B). For automated evaluation, GPT-4-LLM (7B) outperforms Alpaca by 0.2, 0.5, and 0.7 on User-Oriented-Instructions-252 (Wang et al., 2022c), Vicuna-Instructions (Chiang et al., 2023), and Unnatural Instructions (Honovich et al., 2022) datasets, respectively. For human evaluation, regarding aspects including helpfulness, honesty, and harmlessness, GPT-4-LLM outperforms Alpaca by 11.7, 20.9, and 28.6 respectively.

4.7 Claude

Claude is a language model trained by fine-tuning the pre-trained language model on an instruction dataset, aiming to generate helpful and harmless responses. The fine-tuning process consists of two stages: (1) supervised fine-tuning on the instruction dataset. The authors created an instruction dataset by collecting 52K different instructions, paired with responses generated by GPT-4. The fine-tuning process takes approximately eight hours on an 8-card 80GB A100 machine with mixed precision and fully shared data parallelism. (2) optimizing the step-1 model with the proximal policy optimization (Schulman et al., 2017) method. The authors first built a comparison dataset by collecting responses from multiple large language models (e.g., GPT-3 (Brown et al., 2020b)) to the given collection of instructions and then asking GPT-4 (OpenAI, 2023) to rate each response. Using the ratings, a reward model is trained. Then, the fine-tuned model from Step 1 is optimized using the reward model with the proximal policy optimization method.

Claude generates more helpful and harmless responses compared to the backbone model. For automatic evaluations, Claude outperforms GPT-3 by 7% on the RealToxicityPrompts (Gehman et al., 2020) in terms of toxicity. For human evaluations, regarding four different aspects, including following correct instructions, following explicit constraints, fewer hallucinations, and generating appropriate responses, Claude outperforms GPT-3 (Brown et al., 2020b) +10%, +20%, -20%, and +10%. respectively.

4.8 WizardLM

WizardLM (7B) (Xu et al., 2023a) is a language model trained by fine-tuning LLaMA (7B) (Touvron et al., 2023a) on the instruction dataset Evol-Instruct generated by ChatGPT (details see Section 3.7). It is fine-tuned on a subset (with 70K) of Evol-Instruct to enable a fair comparison with Vicuna (Chiang et al., 2023). The fine-tuning process takes approximately 70 hours on 3 epochs based on an 8 V100 GPU with the Deepspeed Zero-3 (Rasley et al., 2020) technique. During inference, the max generation length is 2048.

To evaluate LLMs' performances on complex instructions, the authors collected 218 human-generated instructions from real scenarios (e.g., open-source projects, platforms, and forums), called Evol-Instruct testset.

Evaluations are conducted on the Evol-Instruct testset and Vicuna's testset. For human evaluation, WizardLM outperforms Alpaca (7B) (Taori et al., 2023) and Vicuna (7B) by a large margins, and generates equal or better responses on 67%

https://www.anthropic.com/index/introducing-claude

test samples compared to ChatGPT. Automatic evaluation is conducted by asking GPT-4 to rate LLMs' reponses. Specifically, WizardLM gains performance boosts compared to Alpaca by +6.2%, +5.3% on the Evol-Instruct testset and Vicuna's test sets. WizardLM achieves outperforms Vicuna by +5.8 on the Evol-Instruct testset and +1.7% on the Vicuna's test set.

4.9 ChatGLM2

ChatGLM2 (6B) (Du et al., 2022) is a language model trained by fine-tuning GLM (6B) (Du et al., 2022) on a bilingual dataset that contains both English and Chinese instructions The bilingual instruction dataset contains 1.4T tokens, with a 1:1 ratio of Chinese to English. Instructions in the dataset are sampled from the question-answering and dialogue completion tasks. ChatGLM is initialized with GLM, then trained by the three-step fine-tuning strategy, which is akin to InstructGPT (Ouyang et al., 2022). To better model contextual information across multi-turn conversations, the authors expanded the maximum context length from 1024 to 32K. To reduce GPU memory cost in the fine-tuning stage, the authors employed multi-query attention and causal mask strategies. During inference, ChatGLM2 requires 13GB GPU memory with FP16 and supports conversations up to 8K in length with 6GB GPU memory using the INT4 model quantization technique.

Evaluations are conducted on four English and Chinese benchmarks, including MMLU (English) (Hendrycks et al., 2020), C-Eval (Chinese) (Huang et al., 2023), GSM8K (Math) (Cobbe et al., 2021), and BBH (English) (Suzgun et al., 2022). ChatGLM2 (6B) outperforms GLM (6B) and the baseline model ChatGLM (6B) on all benchmarks. Specifically, ChatGLM2 outperforms GLM by +3.1 on MMLU, +5.0 on C-Eval, +8.6 on GSM8K, and +2.2 on BBH. ChatGLM2 achieves better performances than ChatGLM by +2.1, +1.2, +0.4, +0.8 on MMLU, C-Eval, GSM8K and BBH, respectively.

4.10 LIMA

LIMA (65B) (Zhou et al., 2023) is a large language model trained by fine-tuning LLaMA (65B) (Touvron et al., 2023a) on an instruction dataset, which is constructed based on the proposed superficial alignment hypothesis.

The superficial alignment hypothesis refers to the idea that the knowledge and capabilities of a model are almost acquired in the pre-training stage, while the alignment training (e.g., instruction fine-tuning) teaches models to generate responses under user-preferred formalizations. Based on the superficial alignment hypothesis, the authors claimed that large language models can generate user-satisfied responses by fine-tuning it on a small fraction of instruction data. Therefore, the authors built instruction train/valid/test sets to verify this hypothesis.

Evaluations are conducted on the constructed test set. For human evaluations, LIMA outperforms InstructGPT and Alpaca by 17% and 19%, respectively. Additionally, LIMA achieves comparable results to BARD, Cladue, and GPT-4. For automatic evaluation, which is conducted by asking GPT-4 to rate responses and a higher rate score denotes better performance, LIMA outperforms InstructGPT and Alpaca by 20% and 36%, respectively, achieving comparable results to BARD, while underperforming Claude and GPT-4. Experimental results verify the proposed superficial alignment hypothesis.

4.11 Others

OPT-IML (175B) (Iyer et al., 2022) is a large language model trained by fine-tuning the OPT (175B) (Zhang et al., 2022a) model on the constructed Instruction Meta-Learning (IML) dataset, which consists of over 1500 NLP tasks from 8 publicly available benchmarks such as PromptSource (Bach et al., 2022), FLAN (Longpre et al., 2023), and Super-NaturalInstructions (Wang et al., 2022d). After fine-tuning, OPT-IML outperforms OPT across all benchmarks.

Dolly 2.0 (12B) (Conover et al., 2023a) is initialized with the pre-trained language model Pythia (12B) (Biderman et al., 2023), and fine-tuned on the instruction dataset databricks-dolly-15k, which contains 7 categories of NLP tasks such as text classification and information extraction. After fine-tuning, Dolly 2.0 (12B) outperforms Pythia (12B) on the EleutherAI LLM Evaluation Harness benchmark (Gao et al., 2021) by a large margin, and achieves comparable performances to GPT-NEOX (20B) (Black et al., 2022), which has

https://bard.google.com/
https://www.anthropic.com/index/introducing-claude
https://huggingface.co/datasets/databricks/databricks-dolly-15k

two times more parameters compared to Dolly 2.0 (12B).

Falcon-Instruct (40B) (Almazrouei et al., 2023a) is a large language model trained by fine-tuning Falcon (40B) (Almazrouei et al., 2023b) on an English dialogue dataset, which contains 150 million tokens from the Baize dataset (Xu et al., 2023c), with an additional 5% of the data from the RefinedWeb dataset (Penedo et al., 2023). To reduce memory usage, the authors employed flash attention (Dao et al., 2022) and multi-query techniques. For evaluation, Falcon-Instruct (40B) achieved better performance on the Open LLM Leaderboard (Beeching et al., 2023) compared to the baseline model Falcon (40B), and outperforms the Guanaco (65B), which has more model parameters.

Guanaco (7B) (JosephusCheung, 2021) is a multi-turn dialog language model trained by fine-tuning LLaMA (7B) (Touvron et al., 2023a) on the constructed multilingual dialogue dataset. The multilingual dialogue dataset comes from two sources: Alpaca (Taori et al., 2023), which contains 52K English instruction data pairs; and a multilingual (e.g., Simplified Chinese, Traditional Chinese, Japanese, German) dialogue data, which contains 534K+ multi-turn conversations. After fine-tuning, Guanaco is to generate role-specific responses and continuous responses on a given topic in multi-turn conversations.

Minotaur (15B) is a large language model trained by fine-tuning the Starcoder Plus (15B) (Li et al., 2023f) on open-source instruction datasets including WizardLM (Xu et al., 2023a) and GPTeacher-General-Instruct. For model inference, Minotaur supports a maximum context length of 18K tokens.

Nous-Herme (13B) is a large language model trained by fine-tuning LLaMA (13B) (Touvron et al., 2023a) on an instruction dataset, which contains over 300k instructions, sampled from GPTeacher, CodeAlpaca (Chaudhary, 2023), GPT-4-LLM (Peng et al., 2023), Unnatural Instructions (Honovich et al., 2022), and BiologyPhysicsChemistry subsets in the Camel-AI (Li et al., 2023c). Responses are generated by GPT-4. For evaluations, Nous-Herme (13B) achieves comparable performances to GPT-3.5-turbo on multiple tasks like ARC challenge (Clark et al., 2018) and BoolQ (Clark et al., 2019).

TÜLU (6.7B) (Wang et al., 2023c) is a large language model trained by fine-tuning OPT (6.7B) (Zhang et al., 2022a) on a mixed instruction dataset, which contains FLAN V2 (Longpre et al., 2023), CoT (Wei et al., 2022), Dolly (Conover et al., 2023a), Open Assistant-1, GPT4-Alpaca, Code-Alpaca (Chaudhary, 2023), and ShareGPT. After fine-tuning, TÜLU (6.7B) reaches on average 83% of ChatGPT's performance and 68% of GPT-4's performance.

YuLan-Chat (13B) (YuLan-Chat-Team, 2023) is a language model trained by fine-tuning LLaMA (13B) (Touvron et al., 2023a) on a constructed bilingual dataset, which contains 250,000 Chinese-English instruction pairs. After fine-tuning, YuLan-Chat-13B achieves comparable results to the state-of-the-art open-source model ChatGLM (6B) (Du et al., 2022), and outperforms Vicuna (13B) (Chiang et al., 2023) on the English BBH3K (BBH3K is a subset of BBH benchmark (Srivastava et al., 2022)) dataset.

MOSS (16B) is a bilingual dialogue language model, which aims to engage in multi-turn conversations and utilize various plugins, trained by fine-tuning on dialogue instructions. After fine-tuning, MOSS outperforms the backbone model and generates responses that better align with human preferences.

Airoboros (13B) is a large language model trained by fine-tuning LLAMA (13B) (Touvron et al., 2023a) on the Self-instruct dataset (Wang et al., 2022c). After fine-tuning, Airoboros significantly outperforms LLAMA (13B) (Touvron et al., 2023a) on all benchmarks and achieves highly comparable results to models fine-tuned specifically for certain benchmarks.

UltraLM (13B) (Ding et al., 2023a) is a large language model trained by fine-tuning LLAMA (13B) (Touvron et al., 2023a). For evaluation, UltraLM (13B) outperforms Dolly (12B) (Conover et al., 2023a) and achieves the winning rate up to 98%. Additionally, it surpasses the previous best

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard
https://github.com/teknium1/GPTeacher
https://github.com/teknium1/GPTeacher
https://huggingface.co/datasets/OpenAssistant/oasst1
https://huggingface.co/datasets/vicgalle/alpaca-gpt4
https://sharegpt.com/
https://txsun1997.github.io/blogs/moss.html
https://github.com/jondurbin/airoboros

open-source models (i.e., Vicuna (Chiang et al., 2023) and WizardLM (Xu et al., 2023a)) with winning rates of 9% and 28%, respectively.

5 Multi-modality Instruction Fine-tuning

5.1 Multi-modality Datasets

MUL-TIINSTRUCT (Xu et al., 2022) is a multimodal instruction tuning dataset consisting of 62 diverse multimodal tasks in a unified seq-to-seq format. This dataset covers 10 broad categories and its tasks are derived from 21 existing open-sourced datasets. Each task is equipped with 5 expert-written instructions. For the existing tasks, the authors use the input/output pairs from their available open-source datasets to create instances. While for each new task, the authors create 5k to 5M instances by extracting the necessary information from instances of existing tasks or reformulating them. The MUL-TIINSTRUCT dataset has demonstrated its efficiency in enhancing various transfer learning technique. For example, fine-tuning the OFA model (930M) (Wang et al., 2022a) using various transfer learning strategies such as Mixed Instruction Tuning and Sequential Instruction Tuning on MUL-TIINSTRUCT improve the zero-shot performance across all unseen tasks. On commonsense VQA task, OFA fine-tuned on MUL-TIINSTRUCT achieves 50.60 on RougeL and 31.17 on accuracy, while original OFA achieves 14.97 on RougeL and 0.40 on accuracy.

PMC-VQA (Zhang et al., 2023c) is a large-scale medical visual question-answering dataset that comprises 227k image-question pairs of 149k images, covering various modalities or diseases. The dataset can be used for both open-ended and multiple-choice tasks. The pipeline for generating the PMC-VQA dataset involves collecting image-caption pairs from the PMC-OA (Lin et al., 2023) dataset, using ChatGPT to generate question-answer pairs, and manually verifying a subset of the dataset for quality. The authors propose a generative-based model MedVInT for medical visual understanding by aligning visual information with a large language model. MedVInT pretrained on PMC-VQA achieves state-of-the-art performance and outperforms existing models on VQA-RAD (Lau et al., 2018) and SLAKE (Liu et al., 2021a) benchmarks, with 81.6% accuracy on VQA-RAD and 88.0% accuracy on SLAKE.

LAMM (Yin et al., 2023) is a comprehensive multi-modal instruction tuning dataset for 2D image and 3D point cloud understanding. LAMM contains 186K language-image instruction-response pairs, and 10K language-point cloud instruction-response pairs. The authors collect images and point clouds from publicly available datasets and use the GPT-API and self-instruction methods to generate instructions and responses based on the original labels from these datasets. LAMM-Dataset includes data pairs for commonsense knowledge question answering by incorporating a hierarchical knowledge graph label system from the Bamboo (Zhang et al., 2022b) dataset and the corresponding Wikipedia description. The authors also propose the LAMM-Benchmark, which evaluates existing multi-modal language models (MLLM) on various computer vision tasks. It includes 9 common image tasks and 3 common point cloud tasks, and LAMM-Framework, a primary MLLM training framework that differentiates the encoder, projector, and LLM finetuning blocks for different modalities to avoid modality conflicts.

5.2 Multi-modality Instruction Fine-tuning Models

InstructPix2Pix (983M) (Brooks et al., 2022) is a conditional diffusion model trained by fine-tuning Stable Diffusion (983M) (Rombach et al., 2022) on a constructed multi-modal dataset that contains more than 450K text editing instructions and corresponding images before and after the edit. The authors combine the abilities of two large-scale pre-trained models, a language model GPT-3 (Brown et al., 2020b) and a text-to-image model Stable Diffusion (Rombach et al., 2022), to generate the the training dataset. GPT-3 is fine-tuned to generate text edits based on image prompts, while Stable Diffusion is used to convert the generated text edits into actual image edits. InstructPix2Pix is then trained on this generated dataset using a latent diffusion objective. Figure 5 shows the process of generating image editing dataset and training the diffusion model on that dataset. The authors compares the proposed method qualitatively with previous works such as SDEdit (Meng et al., 2022) and Text2Live (Bar-Tal et al., 2022), highlighting the ability of the model to follow image editing instructions instead of descriptions of the image or

Multi-modality Instruction Fine-tuning Dataset	Modalities		# Tasks
	Modality Pair	# Instance	
MUL-TIINSTRUCT (Xu et al., 2022)[1]	Image-Text	5k to 5M per task	62
PMC-VQA (Zhang et al., 2023c)[2]	Image-Text	227k	2
LAMM (Yin et al., 2023)[3]	Image-Text	186k	9
	Point Cloud-Text	10k	3

[1] https://github.com/VT-NLP/MultiInstruct
[2] https://github.com/xiaoman-zhang/PMC-VQA
[3] https://github.com/OpenLAMM/LAMM

Table 5: An overview of multi-modality instruction fine-tuning datasets.

Multi-modality Instruction Fine-tuned LLMs	# Params	Modality	Base Model		Fine-tuning Trainset	
			Model Name	# Params	Self-build	Size
InstructPix2Pix (Brooks et al., 2022)[1]	983M	I/T	Stable Diffusion	983M	Yes	450K
LLaVA (Liu et al., 2023b)[2]	13B	I/T	CLIP (Radford et al., 2021)	400M	Yes	158K
			LLaMA (Touvron et al., 2023a)	7B		
			LLaMA (Touvron et al., 2023a)	7B		
Video-LLaMA (Zhang et al., 2023b)[3]	-	I/T/V/A	BLIP-2 (Li et al., 2023d)	-	No	-
			ImageBind (Girdhar et al., 2023)	-		
			Vicuna (Chiang et al., 2023)	7B/13B		
InstructBLIP (1.2B) (Dai et al., 2023)[4]	-	I/T/V	BLIP-2 (Li et al., 2023d)	-	No	-
Otter (Li et al., 2023b)[5]	-	I/T/V	OpenFlamingo (Awadalla et al., 2023)	9B	Yes	2.8M
MultiModal-GPT (Gong et al., 2023)[6]	-	I/T/V	OpenFlamingo (Awadalla et al., 2023)	9B	No	-

[1] https://github.com/timothybrooks/instruct-pix2pix
[2] https://github.com/haotian-liu/LLaVA
[3] https://github.com/DAMO-NLP-SG/Video-LLaMA
[4] https://github.com/salesforce/LAVIS/tree/main/projects/instructblip
[5] https://github.com/Luodian/Otter
[6] https://github.com/open-mmlab/Multimodal-GPT

Table 6: An overview of multi-modality instruction fine-tuned LLMs. I/T/V/A stand for Image/Text/Video/Audio

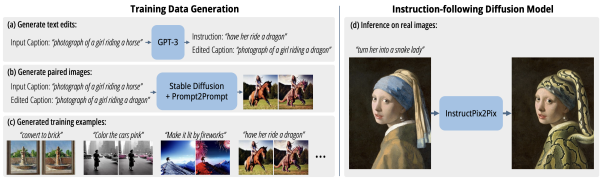


Figure 5: Image editing dataset generation and diffusion model training. The figure is copied from Brooks et al. (2022).

edit layer. The authors also presents quantitative comparisons with SDEdit (Meng et al., 2022) using metrics measuring image consistency and edit quality.

LLaVA (13B) (Liu et al., 2023b) is a large multimodal model developed by connecting the visual encoder of CLIP (400M) (Radford et al., 2021) with the language decoder LLaMA (7B) (Touvron et al., 2023a). LLaVA is fine-tuned using the generated instructional vision-language dataset consisted of 158K unique language-image instruction-following samples. The data collection process involved creating conversation, detailed description, and complex reasoning prompts. GPT-4 is used to convert image-text pairs into appropriate instruction-following format for this dataset. Visual features such as captions and bounding boxes were used to encode images. LLaVA yields a 85.1% relative score compared with GPT-4 on a synthetic multimodal instruction following dataset. When fine-tuned on Science QA, the synergy of LLaVA and GPT-4 achieves a new state-of-the-art accuracy of 92.53%.

Video-LLaMA (Zhang et al., 2023b) is a multimodal framework that enhances large language models with the ability to understand

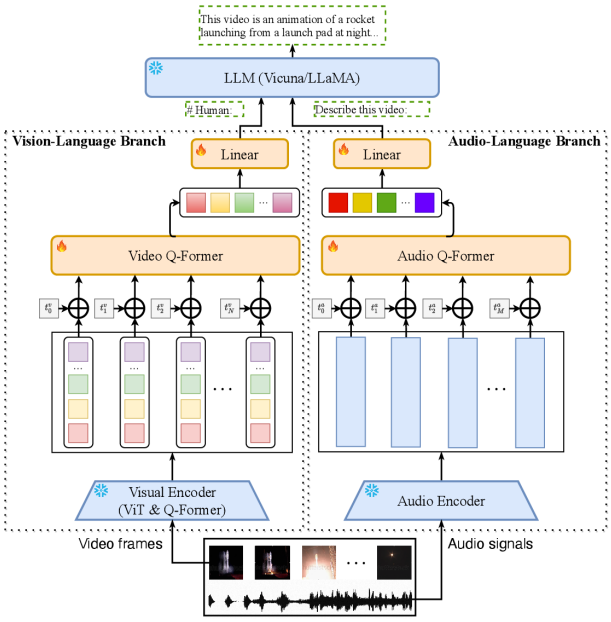


Figure 6: Overall architecture of Video-LLaMA. The figure is copied from Zhang et al. (2023b).

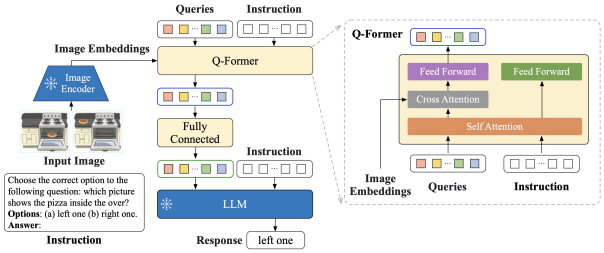


Figure 7: Overall architecture of InstructBLIP. The figure is copied from Dai et al. (2023).

both visual and auditory content in videos. The architecture of Video-LLaMA consists of two branche encoders: the Vision-Language (VL) Branch and the Audio-Language (AL) Branch, and a language decoder (Vicuna (7B/13B) (Chiang et al., 2023), LLaMA (7B) (Touvron et al., 2023a), etc.). The VL Branch includes a frozen pre-trained image encoder (pre-trained vision component of BLIP-2 (Li et al., 2023d), which includes a ViT-G/14 and a pre-trained Q-former), a position embedding layer, a video Q-former and a linear layer. The AL Branch includes a pre-trained audio encoder (ImageBind (Girdhar et al., 2023)) and an Audio Q-former. Figure 6 shows the overall architecture of Video-LLaMA with Vision-Language Branch and Audio-Language Branch. The VL Branch is trained on the Webvid-2M (Bain et al., 2021) video caption dataset with a video-to-text generation task, and fine-tuned on the instruction-tuning data from MiniGPT-4 (Zhu et al., 2023), LLaVA (Liu et al., 2023b) and VideoChat (Li et al., 2023e). The AL Branch is trained on video/image instru-caption data to connect the output of ImageBind to language decoder. After finetuning, Video-LLaMA can perceive and comprehend video content, demonstrating its ability to integrate auditory and visual information, understand static images, recognize common-knowledge concepts, and capture temporal dynamics in videos.

InstructBLIP (1.2B) (Dai et al., 2023) is a vision-language instruction tuning framework initialized with a pre-trained BLIP-2 (Li et al., 2023d)) model consisting of an image encoder, an LLM (FlanT5 (3B/11B) (Chung et al., 2022) or Vicuna (7B/13B) (Chiang et al., 2023)), and a Query Transformer (Q-Former) to bridge the two. As shown in Figure 7, the Q-Former extracts instruction-aware visual features from the output embeddings of the frozen image encoder, and feeds the visual features as soft prompt input to the frozen LLM. The authors evaluate the proposed InstructBLIP model on a variety of vision-language tasks, including image classification, image captioning, image question answering, and visual reasoning. They use 26 publicly available datasets, dividing them into 13 held-in and 13 held-out datasets for training and evaluation. The authors demonstrate that InstructBLIP achieves state-of-the-art zero-shot performance on a wide range of vision-language tasks. InstructBLIP yields an average relative improvement of 15.0% when compared to BLIP-2, smallest InstructBLIP (4B) outperforms Flamingo (80B) (Alayrac et al., 2022) on all six shared evaluation datasets with an average relative improvement of 24.8%.

Otter (Li et al., 2023b) is a multi-modal model trained by fine-tuning OpenFlamingo (9B) (Awadalla et al., 2023), with the language and vision encoders frozen and only fine-tuning the Perceiver resampler module, cross-attention layers, and input/output embeddings. The authors organize diverse multi-modal tasks covering 11 categories and build multi-modal in-context instruction tuning datasets MIMIC-IT of 2.8M multimodal instruction-response pairs, which consists of image-instruction-answer triplets, where the instruction-answer is tailored to the image. Each data sample also includes context, which contains a series of image-instruction-answer triplets that

contextually correlate with the queried triplet. Otter demonstrates the ability to follow user instructions more accurately and provide more detailed descriptions of images compared to OpenFlamingo (Awadalla et al., 2023).

MultiModal-GPT (Gong et al., 2023) is a multi-modal instruction tuning model that is capable of following diverse instructions, generating detailed captions, counting specific objects, and addressing general inquiries. MultiModal-GPT is trained by fine-tuning OpenFlamingo (9B) (Awadalla et al., 2023) on various created visual instruction data with open datasets, including VQA, Image Captioning, Visual Reasoning, Text OCR, and Visual Dialogue. The experiments demonstrate the proficiency of MultiModal-GPT in maintaining continuous dialogues with humans.

6 Domain-specific Instruction Finetuning

In this section, we describe instruction tuning in different domains and applications.

6.1 Dialogue

InstructDial (Gupta et al., 2022) is an instruction tuning framework designed for dialogue. It contains a collection of 48 dialogue tasks in a consistent text-to-text format created from 59 dialogue datasets. Each task instance includes a task description, instance inputs, constraints, instructions, and output. To ensure adherence to instructions, the framework introduces two meta-tasks: (1) an instruction selection task, where the model selects the instruction corresponding to a given input-output pair; and (2) an instruction binary task, where the model predicts "yes" or "no" if an instruction leads to a given output from an input. Two base models T0-3B (Sanh et al., 2021) (3B parameters version of T5 (Lester et al., 2021)) and BART0 (Lin et al., 2022) (406M parameters based on Bart-large (Lewis et al., 2019)) are fine-tuned on the tasks from InstructDial. InstructDial achieves impressive results on unseen dialogue datasets and tasks, including dialogue evaluation and intent detection. Moreover, it delivers even better results when applied to a few-shot setting.

6.2 Intent Classification and Slot Tagging

LINGUIST (Rosenbaum et al., 2022) finetunes AlexaTM 5B (Soltan et al., 2022), a 5-billion-parameter multilingual model, on the instruction dataset for intent classification and slot tagging

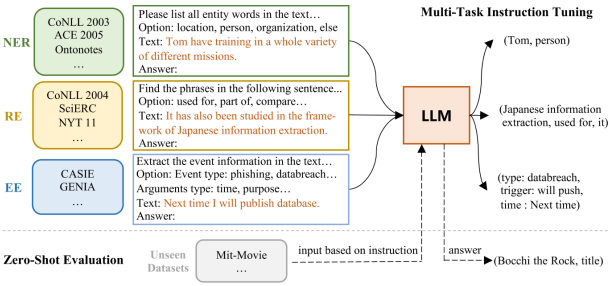


Figure 8: The overview framework of InstructUIE. The figure is copied from Wang et al. (2023b).

tasks. Each instruction consists of five blocks: (i) the language of the generated output, (ii) intention, (iii) slot types and values to include in the output (e.g., the number 3 in [3, snow] corresponds the slot type, and snow is the value used for that slot), (iv) a mapping from slot type labels to numbers, and (v) up to 10 examples to instruct the format of the outputs. LINGUIST shows significant improvements over state-of-the-art approaches in a 10-shot novel intent setting using the SNIPS dataset (Coucke et al., 2018). In the zero-shot cross-lingual setting of the mATIS++ dataset (Xu et al., 2020), LINGUIST surpasses a strong baseline of Machine Translation with Slot Alignment across 6 languages while maintaining intent classification performance.

6.3 Information Extraction

InstructUIE (Wang et al., 2023b) is a unified information extraction (IE) framework based on instruction tuning, which transforms IE tasks to the seq2seq format and solves them by fine-tuning 11B FlanT5 (Chung et al., 2022) on the constructed IT dataset. Figure 8 shows the overall architecture of InstructUIE. It introduces IE INSTRUCTIONS, a benchmark of 32 diverse information extraction datasets in a unified text-to-text format with expert-written instructions. Each task instance is delineated by four properties: task instruction, options, text, and output. Task instruction contains information such as the type of information to be extracted, the output structure format, and additional constraints or rules that need to be adhered to during the extraction process. Options refer to the output label constraints of a task. Text refers to the input sentence. Output is the sentence obtained by converting the original tags of the sample (e.g. "entity tag: entity span" for NER). In the supervised setting, InstructUIE performs comparably to BERT (Devlin et al.,

2018) and outperforms the state-of-the-art and GPT3.5 (Brown et al., 2020a) in zero-shot settings.

6.4 Aspect-based Sentiment Analysis

Varia et al. (2022) propose a unified instruction tuning framework for solving Aspect-based Sentiment Analysis (ABSA) task based on a fine-tuned T5 (220M) (Raffel et al., 2019) model. The framework addresses multiple factorized sub-tasks that involve the four elements of ABSA, namely Aspect Term, Aspect Category, Opinion Term, and Sentiment. It treats these sub-tasks as a combination of five Question Answering (QA) tasks by transforming each sentence in the corpus using instruction templates provided for each task. For instance, one of the instruction templates used is "What are the aspect terms in the text: $TEXT?". The framework showcases substantial improvement (8.29 F1 on average) over the state-of-the-art in few-shot learning scenarios and remains comparable in full fine-tuning scenarios.

6.5 Writing

Zhang et al. (2023d) propose Writing-Alpaca-7B that fine-tunes LLaMa-7B on the writing instruction dataset to provide writing assistance. The proposed instruction dataset is an extension of the EDITEVAL benchmark based on instructional data, with the Updating task removed and a task for grammaticality introduced. The instruction scheme strictly follows the one in the Stanford Alpaca project, comprising a universal preface, an instruction field to guide task completion, an input field that provides the text to be edited, and a response field that requires models to fill out. The Writing-Alpaca-7B improves upon LLaMa's performance on all writing tasks and outperforms other larger off-the-shelf LLMs.

CoEdIT (Raheja et al., 2023) finetunes FLANT5 (770M parameters, 3B parameters, and 11B parameters) on the instruction dataset for text editing to provide writing assistance. The instruction dataset comprises approximately 82K <instruction: source, target> pairs. As shown in Figure 9, the model takes instructions from the user specifying the characteristics of the desired text, such as "Make the sentence simpler", and outputs the edited text. CoEdIT achieves state-of-the-art performance on several text editing tasks, including grammatical error correction, text simplification, iterative text editing, and three stylistic editing tasks: formality style transfer, neutralization, and paraphrasing. Furthermore, it can generalize well to new, adjacent tasks not seen during fine-tuning.

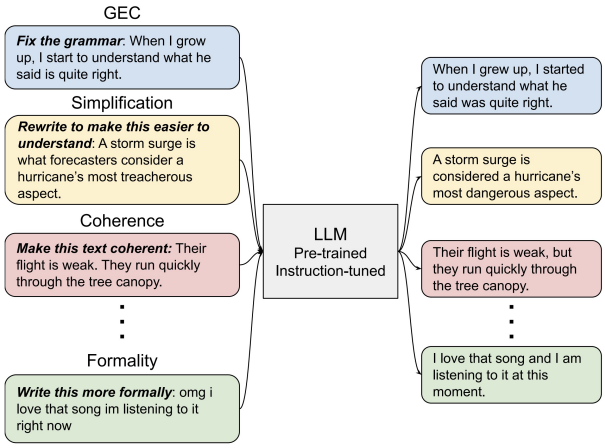


Figure 9: The overview framework of COEDIT. The figure is copied from Raheja et al. (2023).

CoPoet (Chakrabarty et al., 2022) is a collaborative poetry writing tool that utilizes a large language model (e.g. T5-3B, T5-11B and T0-3B models) trained on a diverse collection of instructions for poetry writing. Each sample in the instruction dataset includes an <instruction, poem_line> pair. There are three major types of instructions: Continuation, Lexical Constraints, and Rhetorical Techniques. The CoPoet is guided by user instructions that specify desired attributes of the poetry, such as writing a sentence about "love" or ending a sentence with "fly." Not only is the system competitive with publicly available LLMs trained on instructions, such as InstructGPT, but it is also capable of satisfying unseen compositional instructions.

6.6 Medical

Radiology-GPT (Liu et al., 2023c) is a fine-tuned Alpaca-7B model for radiology, which utilizes an instruction tuning approach on an extensive dataset of radiology domain knowledge. Radiology reports usually include two corresponding sections: "Findings" and "Impression". The "Findings" section contains detailed observations from the radiology images, while the "Impression" section summarizes the interpretations drawn from those observations. Radiology-GPT provides a brief instruction to the "Findings" text: "Derive the impression from findings in the radiology report". The "Impression" text from the same report serves

as the target output. In comparison to general language models such as StableLM, Dolly, and LLaMA, Radiology-GPT demonstrates significant versatility in radiological diagnosis, research, and communication.

ChatDoctor (Li et al., 2023g) is based on the fine-tuned LLaMA-7B model, utilizing the alpaca instruction dataset and the HealthCareMagic100k patient-doctor dialogue dataset. And prompt templates are designed for retrieving external knowledge databases, such as the Disease Database and Wikipedia retrieval, during doctor-patient conversations to obtain more accurate outputs from the model. The ChatDoctor significantly improves the model'sability to comprehend patient needs and provide informed advice. By equipping the model with self-directed information retrieval from reliable online and offline sources, the accuracy of its responses is substantially improved.

ChatGLM-Med (Haochun Wang, 2023) is fine-tuned on the Chinese medical instruction dataset based on the ChatGLM-6B model. The instruction dataset comprises medically relevant question and answer pairs, created using the GPT3.5 API and the Medical Knowledge Graph. This model improves the question-answering performance of ChatGLM in the medical field.

6.7 Arithmetic

Goat (Liu and Low, 2023) is a fine-tuned LLaMA-7B model based on instructions, which aims to solve arithmetic problems. It expresses arithmetic problems in the form of natural language question answering, such as "What is 8914/64?", by generating hundreds of instruction templates using ChatGPT. The model applies various techniques to enhance its adaptability to diverse question formats, such as randomly removing spaces between numbers and symbols in the arithmetic expression and replacing "*" with "x" or "times". The Goat model achieves state-of-the-art performance on the BIG-bench arithmetic subtask. In particular, zero-shot Goat7B matches or exceeds the accuracy achieved by the few-shot PaLM-540B.

6.8 Code

WizardCoder (Luo et al., 2023) utilizes StarCoder 15B as the foundation with complex instruction fine-tuning, by adapting the Evol-Instruct method (Xu et al., 2023) to the domain of code. The training dataset is produced through iterative application of the Evol-Instruct technique on the Code Alpaca dataset, which includes the following attributes for each sample: instruction, input, and expected output. For instance, when the instruction is "Amend the following SQL query to select distinct elements", the input is the SQL query, and the expected output is the generated answer. The WizardCoder outperforms all other open-source Code LLMs and even outperforms the largest LLMs, Anthropic's Claude and Google's Bard, on HumanEval and HumanEval+.

7 Efficient Tuning Techniques

Efficient fine-tuning techniques aim at adapting LLMs to downstream tasks by optimizing a small fraction of parameters in multiple ways, *i.e.*, addition-based, specification-based, and reparameterization-based. Addition-based methods introduce extra trainable parameters or modules not present in the original model. Representative methods include adapter tuning (Houlsby et al., 2019) and prompt-based tuning (Schick and Schütze, 2021). Specification-based methods specify certain inherent model parameters to be tuned while freezing others. For example, BitFit (Zaken et al., 2022) tunes the bias terms of the pre-trained model. Reparameterization methods transform model weights into more parameter-efficient forms for tuning. The key hypothesis is that model adaptation is low-rank, so weights can be reparameterized into low-rank factors or a low-dimensional subspace (*e.g.*, LoRA (Hu et al., 2021)). Intrinsic prompt tuning finds a low-dimensional subspace shared by tuning prompts across diverse tasks.

7.1 LoRA

Low-Rank Adaptation (LoRA) (Hu et al., 2021) enables efficient adaptation of LLMs using low-rank updates. LoRA use DeepSpeed (Rasley et al., 2020) as the training backbone. The key insight of LoRA is that the actual change in LLMs' weights required for new task adaptation lies in a low-dimensional subspace. Specifically, for a pretrained weight matrix W_0, the authors model the adapted weight matrix as $W_0 + \Delta W$, where ΔW is a low rank update. ΔW is parameterized as $\Delta W = BA$, where A and B are much smaller trainable matrices. The rank r of ΔW is chosen to be much smaller than the dimensions of W_0. The intuition is that instead of directly training all of W_0, the authors

train low-dimensional A and B, which indirectly trains W_0 in a low-rank subspace of directions that matter for the downstream task. This results in far fewer trainable parameters compared to full fine-tuning. For GPT-3, LoRA reduces the number of trainable parameters by 10,000x and memory usage by 3x compared to full fine-tuning.

7.2 HINT

HINT (Ivison et al., 2022) combines the generalization benefits of instruction tuning with efficient on-demand fine-tuning, avoiding repeatedly processing lengthy instructions. The essence of HINT lies in hypernetworks, which generate parameter-efficient modules for LLMs adaptation based on natural language instructions and few-shot examples. The adopted hypernetwork converts instructions and few-shot examples into a encoded instruction and generates adapter and prefix parameters using a pretrained text encoder and cross-attention based parameter generator. Then, the generated adapters and prefixes are inserted into the backbone model as efficient tuning modules. At inference, the hypernetwork performs inference only once per task to generate adapted modules. The benefits are that HINT can incorporate long instructions and additional few-shots without increasing compute, unlike regular fine-tuning or input concatenation methods.

7.3 Qlora

QLORA (Dettmers et al., 2023) includes optimal quantization and memory optimization, aiming at providing efficient and effective LLMs fine-tuning. QLORA includes 4-bit NormalFloat (NF4) Quantization, which is a quantization scheme optimized for the typical normal distribution of LLM weights. By quantizing based on the quantiles of a normal distribution, NF4 provides better performance than standard 4-bit integer or float quantization. To further reduce memory, the quantization constants are themselves quantized to 8 bits. This second level of quantization saves an additional 0.37 bits per parameter on average. QLORA leverages NVIDIA's unified memory feature to page optimizer states to CPU RAM when GPU memory is exceeded. avoiding out-of-memory during training. QLORA enables training a 65B parameter LLM on a single 48GB GPU with no degradation compared to full 16-bit finetuning. QLORA works by freezing the 4-bit quantized base LLM, then backpropagating through it into a small set of 16-bit low-rank adapter weights which are learned.

7.4 LOMO

LOw-Memory Optimization (LOMO) (Lv et al., 2023) enables full parameter fine-tuning of LLMs using limited computational resources through a fusion of gradient computation and update. The essence is to fuse gradient computation and parameter update into one step during backpropagation, thereby avoiding storage of full gradient tensors. Firstly, theoretical analysis is provided in LOMO on why SGD can work well for fine-tuning large pre-trained models despite its challenges on smaller models. In addition, LOMO updates each parameter tensor immediately after computing its gradient in backpropagation. Storing the gradient of one parameter at a time reduces gradient memory to $O(1)$. LOMO employs gradient value clipping, separate gradient norm computation pass and dynamic loss scaling to stabilize training. The integration of activation checkpointing and ZeRO optimization methods saves memory.

7.5 Delta-tuning

Delta-tuning (Ding et al., 2023b) provides optimization and optimal control perspectives for theoretical analyzation. Intuitively, delta-tuning performs subspace optimization by restricting tuning to a low-dimensional manifold. The tuned parameters act as optimal controllers guiding model behavior on downstream tasks.

8 Evaluation, Analysis and Criticism

8.1 HELM Evaluation

HELM(Liang et al., 2022) is a holistic evaluation of Language Models (LMs) to improve the transparency of language models, providing a more comprehensive understanding of the capabilities, risks, and limitations of language models. Specifically, differing from other evaluation methods, HELM holds that a holistic evaluation of language models should focus on the following three factors:

(1) **Broad coverage.** During the development, language models can be adapted to various NLP tasks (e.g., sequence labeling and question answering), thus, the evaluation of language models needs to be carried out in a wide range of scenarios. To involve all potential scenarios,

HELM proposed a top-down taxonomy, which begins by compiling all existing tasks in a major NLP conference (ACL2022) into a task space and dividing each task into the form of scenarios (e.g., languages) and metrics (e.g., accuracy). Then when facing a specific task, the taxonomy would select one or more scenarios and metrics in the task space to cover it. By analyzing the structure of each task, HELM clarifies the evaluation content (task scenarios and metrics) and improves the scenario coverage of language models from 17.9% to 96.0%.

(2) **Multi-metric measurement.** In order to enable human to weigh language models from different perspectives, HELM proposes multi-metric measurement. HELM has covered 16 different scenarios and 7 metrics. To ensure the results of intensive multi-metric measurement, HELM measured 98 of 112 possible core scenarios (87.5%).

(3) **Standardization.** The increase in the scale and training complexity of language models has seriously hindered human's understanding of the structure of each language model. To establish a unified understanding of existing language models, HELM benchmarks 30 well-known language models, covering such institutions as Google (UL2(Tay et al., 2022)), OpenAI (GPT-3(Brown et al., 2020b)), and EleutherAI (GPT-NeoX(Black et al., 2022)). Interestingly, HELM pointed out that LMs such as T5 (Raffel et al., 2019) and Anthropic-LMv4-s3 (Bai et al., 2022a) had not been directly compared in the initial work, while LLMs such as GPT-3 and YaLM were still different from their corresponding reports after multiple evaluations.

8.2 Low-resource Instruction Tuning

Gupta et al. (2023) attempts to estimate the minimal downstream training data required by IT models to match the SOTA supervised models over various tasks. Gupta et al. (2023) conducted experiments on 119 tasks from Super Natural Instructions (SuperNI) in both single-task learning (STL) and multi-task learning (MTL) settings. The results indicate that in the STL setting, IT models with only 25% of downstream training data outperform the SOTA models on those tasks, while in the MTL setting, just 6% of downstream training data can lead IT models to achieve the SOTA performance. These findings suggest that instruction tuning can effectively assist a model in quickly learning a task even with limited data.

However, due to resource limitations, Gupta et al. (2023) did not conduct experiments on LLMs, like T5-11B. So, to gain a more comprehensive understanding of the IT models, further investigation using larger language models and datasets is necessary.

8.3 Smaller Instruction Dataset

IT requires a substantial amount of specialized instruction data for training. Zhou et al. (2023) hypothesized that the pre-trained LLM only has to learn the style or format to interact with users and proposed LIMA that achieves strong performance by fine-tuning an LLM on only 1,000 carefully selected training examples.

Specifically, LIMA first manually curates 1,000 demonstrations with high-quality prompts and responses. Then the 1,000 demonstrations are used to fine-tune the pre-trained 65B-parameter LLaMa (Touvron et al., 2023b). By comparison, across more than 300 challenging tasks, LIMA outperfrms GPT-davinci003 (Brown et al., 2020b), which was fine-tuned on 5,200 examples by human feedback tuning. Moreover, with only half amount of demonstrations, LIMA achieves equivalent results to GPT-4 (OpenAI, 2023), Claude (Bai et al., 2022b), and Bard. Above all, LIMA demonstrated that LLMs' powerful knowledge and capabilities can be exposed to users with only a few carefully curated instructions to fine-tune.

8.4 Evaluating Instruction-tuning Datasets

The performance of IT model highly depends on the IT datasets. However, there lacks of evaluations for these IT datasets from open-ended and subjective aspects.

To address this issue, Wang et al. (2023c) performs dataset evaluation by fine-tuning the LLaMa model (Touvron et al., 2023b) on various of open IT datasets and measure different fine-tuned models through both automatic and human evaluations. An additional model is trained on the combination of IT datasets. For the results, Wang et al. (2023c) showed that there is not a single best IT dataset across all tasks, while by manually combining datasets it can achieve the best overall performance. Besides, Wang et al. (2023c) pointed out that though IT can bring large benefits on LLMs at all sizes, smaller models and models

Bard, designed by Google, is an interface to generative AI platform, and the link is: https://ai.google/static/documents/google-about-bard.pdf

with a high base quality benefit most from IT. For human evaluations, Wang et al. (2023c) a larger model is more likely to gain a higher acceptability score.

8.5 Do IT just learn Pattern Copying?

To address the lack of clarity about the specific knowledge that models acquire through instruction tuning, Kung and Peng (2023) delves into the analysis of how models make use of instructions during IT by comparing the tuning when provided with altered instructions versus the original instructions.

Specifically, Kung and Peng (2023) creates simplified task definitions that remove all semantic components, leaving only the output information. In addition, Kung and Peng (2023) also incorporates delusive examples that contain incorrect input-output mapping. Surprisingly, the experiments show that models trained on these simplified task definitions or delusive examples can achieve comparable performance to the ones trained on the original instructions and examples. Moreover, the paper also introduces a baseline for the classification task with zero-shot, which achieves similar performance to IT in low-resource settings.

In summary, according to Kung and Peng (2023), the notable performance improvements observed in current IT models may be attributed to their ability to capture surface-level patterns, such as learning the output format and making guesses, rather than comprehending and learning the specific task.

8.6 Proprietary LLMs Imitation

LLMs imitation is an approach that collects outputs from a stronger model, such as a proprietary system like ChatGPT, and uses these outputs to fine-tune an open-source LLM. Through this way, an open-source LLM may get competitive capabilities with any proprietary model.

Gudibande et al. (2023) conducted several experiments to critically analyze the efficacy of model imitation. Specifically, Gudibande et al. (2023) first collected datasets from outputs of ChatGPT over broad tasks. Then these datasets were used to fine-tune a range of models covering sizes from 1.5B to 13B, base models GPT-2 and LLaMA, and data amounts from 0.3M tokens to 150M tokens.

For evaluations, Gudibande et al. (2023) demonstrated that on tasks with supported datasets, imitation models are far better than before, and their outputs appear similar to ChatGPT's. While on tasks without imitation datasets, imitation models do not have improvement or even decline in accuracy.

Thus, Gudibande et al. (2023) pointed out that it's the phenomenon that imitation models are adept at mimicking ChatGPT's style (e.g., being fluent, confident and well-structured) that makes researchers have the illusion about general abilities of imitation models. So, Gudibande et al. (2023) suggested that instead of imitating proprietary models, researchers had better focus on improving the quality of base models and instruction examples.

9 Conclusion

This work surveys recent advances in the fast growing field of instruction tuning. We make a systematic review of the literature, including the general methodology of IT, the construction of IT datasets, the training of IT models, IT's applications to different modalities, domains and application. We also review analysis on IT models to discover both their advantages and potential pitfalls. We hope this work will act as a stimulus to motivate further endeavors to address the deficiencies of current IT models.

References

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2023. Evaluating correctness and faithfulness of instruction-following models for question answering. *ArXiv*, abs/2307.16877.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *NeurIPS*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023a. Falcon-40B: an open large language model with state-of-the-art performance.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, et al. 2023b. Falcon-40b: an open large language model with state-of-the-art performance.

Anas Awadalla, Irena Gao, Joshua Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Jenia Jitsev, et al. 2023. Openflamingo.

Stephen H. Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Févry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Mike Tian-Jian Jiang, and Alexander M. Rush. 2022. Promptsource: An integrated development environment and repository for natural language prompts. *ArXiv*, abs/2202.01279.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*.

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. 2022. Text2live: Text-driven layered image and video editing. In *European Conference on Computer Vision*, pages 707–723. Springer.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *International Conference on Web and Social Media*.

Edward Beeching, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. *Hugging Face*.

Stella Rose Biderman, Hailey Schoelkopf, Quentin G. Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *ArXiv*, abs/2304.01373.

Sid Black, Stella Rose Biderman, Eric Hallahan, Quentin G. Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Martin Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Benqi Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *ArXiv*, abs/2204.06745.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. 2022. Instructpix2pix: Learning to follow image editing instructions. *ArXiv*, abs/2211.09800.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Tuhin Chakrabarty, Vishakh Padmakumar, and Hengxing He. 2022. Help me write a poem - instruction tuning as a vehicle for collaborative poetry writing. *ArXiv*, abs/2210.13669.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Arun Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Qianglong Chen, Guohai Xu, Mingshi Yan, Ji Zhang, Fei Huang, Luo Si, and Yin Zhang. 2023. Distinguish before answer: Generating contrastive explanation as knowledge for commonsense question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *ArXiv*, abs/1905.10044.

J. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

OpenAccess AI Collective. 2023. *software: huggingface.co/openaccess-ai-collective/minotaur-15b*.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, et al. 2023a. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023b. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv*, abs/2305.06500.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023a. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Ning Ding, Yujia Qin, Guang Yang, Fu Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Haitao Zheng, Jianfei Chen, Y. Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023b. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5:220–235.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Jon Durbin. 2023. Airoboros. *software: github.com/jondurbin/airoboros*.

William Fedus, Barret Zoph, and Noam M. Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023a. Exploring the feasibility of chatgpt for event extraction. *ArXiv*, abs/2303.03836.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *ArXiv*, abs/2009.11462.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *CVPR*.

Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qianmengke Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. 2023. Multimodal-gpt: A vision and language model for dialogue with humans. *ArXiv*, abs/2305.04790.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*.

Himanshu Gupta, Saurabh Arjun Sawant, Swaroop Mishra, Mutsumi Nakamura, Arindam Mitra, Santosh Mashetty, and Chitta Baral. 2023. Instruction tuned models are quick learners. *arXiv preprint arXiv:2306.05539*.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskénazi, and Jeffrey P. Bigham. 2022. Instructdial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Conference on Empirical Methods in Natural Language Processing*.

Sendong Zhao Bing Qin Ting Liu Haochun Wang, Chi Liu. 2023. Chatglm-med. https://github.com/SCIR-HI/Med-ChatGLM.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Hamish Ivison, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew E. Peters. 2022. Hint: Hypernetwork instruction tuning for efficient zero-shot generalisation. *ArXiv*, abs/2212.10315.

Srinivas Iyer, Xiaojuan Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Veselin Stoyanov. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *ArXiv*, abs/2212.12017.

JosephusCheung. 2021. Guanaco: Generative universal assistant for natural-language adaptive context-aware omnilingual outputs.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. *ArXiv*, abs/2305.11383.

LAION.ai. 2023. Oig: the open instruction generalist dataset.

Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10.

Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023a. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *ArXiv*, abs/2304.11633.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023b. Otter: A multi-modal model with in-context instruction tuning. *ArXiv*, abs/2305.03726.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023c. Camel: Communicative agents for "mind" exploration of large scale language model society.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023d. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023e. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023f. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, and You Zhang. 2023g. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *ArXiv*, abs/2303.14070.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher R'e, Diana Acosta-Navas, Drew A. Hudson, E. Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan S. Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas F. Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*.

Bill Yuchen Lin, Kangmin Tan, Chris Miller, Beiwen Tian, and Xiang Ren. 2022. Unsupervised cross-task generalization via retrieval augmentation. *ArXiv*, abs/2204.07937.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. In *Annual Meeting of the Association for Computational Linguistics*.

Weixiong Lin, Ziheng Zhao, Xiaoman Zhang, Chaoyi Wu, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-clip: Contrastive language-image pre-training using biomedical documents. *arXiv preprint arXiv:2303.07240*.

Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. 2021a. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1650–1654. IEEE.

Hanmeng Liu, Zhiyang Teng, Leyang Cui, Chaoli Zhang, Qiji Zhou, and Yue Zhang. 2023a. Logicot: Logical chain-of-thought instruction-tuning data collection with gpt-4. *ArXiv*, abs/2305.12147.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *ArXiv*, abs/2304.08485.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021b. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*.

Zheng Liu, Aoxiao Zhong, Yiwei Li, Longtao Yang, Chao Ju, Zihao Wu, Chong Ma, Peng Shu, Cheng Chen, Sekeun Kim, Haixing Dai, Lin Zhao, Dajiang Zhu, Jun Liu, Wei Liu, Dinggang Shen, Xiang Li, Quanzheng Li, and Tianming Liu. 2023c. Radiology-gpt: A large language model for radiology.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct.

Kai Lv, Yuqing Yang, Tengxiao Liu, Qi jie Gao, Qipeng Guo, and Xipeng Qiu. 2023. Full parameter fine-tuning for large language models with limited resources.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Munan Ning, Yujia Xie, Dongdong Chen, Zeyin Song, Lu Yuan, Yonghong Tian, Qixiang Ye, and Liuliang Yuan. 2023. Album storytelling with iterative story-aware captioning and large language models. *ArXiv*, abs/2305.12943.

NousResearch. 2023. *software: huggingface.co/NousResearch/Nous-Hermes-13b*.

OpenAI. 2022. Introducing chatgpt. *Blog post openai.com/blog/chatgpt*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. *arXiv preprint arXiv:2202.13257*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. Coedit: Text editing by task-specific instruction tuning. *ArXiv*, abs/2305.09857.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

Andrew Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. 2022. Linguist: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. In *International Conference on Computational Linguistics*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Teven Le Scao, Angela Fan, Christopher Akiki, Elizabeth-Jane Pavlick, Suzana Ili'c, Daniel Hesslow, Roman Castagn'e, Alexandra Sasha Luccioni, Franccois Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Rose Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurenccon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa Etxabe, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris C. Emezue, Christopher Klamm, Colin Leong, Daniel Alexander van Strien, David Ifeoluwa Adelani, Dragomir R. Radev, Eduardo Gonz'alez Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady ElSahar, Hamza Benyamina, Hieu Trung Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jorg Frohberg, Josephine L. Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro von Werra, Leon Weber, Long Phan, Loubna Ben Allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, Mar'ia Grandury, Mario vSavsko, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad Ali Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla A. Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto L'opez, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, S. Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal V. Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Févry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiang Tang, Zheng Xin Yong, Zhiqing Sun, Shaked Brody, Y Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre Franccois Lavall'ee, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aur'elie N'ev'eol, Charles Lovering, Daniel H Garrette, Deepak R. Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Xiangru Tang, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, S. Osher Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdenvek Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ananda Santa Rosa Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Olusola Ajibade, Bharat Kumar Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David M. Lansky, Davis David, Douwe Kiela, Duong Anh Nguyen, Edward Tan, Emily Baylor, Ezinwanne Ozoani, Fatim T Mirza, Frankline Ononiwu, Habib Rezanejad, H.A. Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jan Passmore, Joshua Seltzer, Julio Bonis Sanz, Karen Fort, Lívia Macedo Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, M. K. K. Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nourhan Fahmy, Olanrewaju Samuel, Ran An, R. P. Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas L. Wang, Sourav Roy, Sylvain Viguier, Thanh-Cong Le, Tobi Oyebade, Trieu

Nguyen Hai Le, Yoyo Yang, Zachary Kyle Nguyen, Abhinav Ramesh Kashyap, A. Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Kumar Singh, Benjamin Beilharz, Bo Wang, Caio Matheus Fonseca de Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel Le'on Perin'an, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Iman I.B. Bello, Isha Dash, Ji Soo Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthi Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, María Andrea Castillo, Marianna Nezhurina, Mario Sanger, Matthias Samwald, Michael Cullan, Michael Weinberg, M Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patricia Haller, R. Chandrasekhar, R. Eisenberg, Robert Martin, Rodrigo L. Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Pratap Bharati, T. A. Laud, Th'eo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yashasvi Bajaj, Y. Venkatraman, Yifan Xu, Ying Xu, Yun chao Xu, Zhee Xao Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. Bloom: A 176b-parameter open-access multilingual language model. *ArXiv*, abs/2211.05100.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *ArXiv*, abs/2210.03057.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Weiwei Sun, Hengyi Cai, Hongshen Chen, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2023a. Answering ambiguous questions via iterative prompting. *ArXiv*, abs/2307.03897.

Xiaofei Sun, Linfeng Dong, Xiaoya Li, Zhen Wan, Shuhe Wang, Tianwei Zhang, Jiwei Li, Fei Cheng, Lingjuan Lyu, Fei Wu, et al. 2023b. Pushing the limits of chatgpt on nlp tasks. *arXiv preprint arXiv:2306.09719*.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023c. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed Huai hsin Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *ArXiv*, abs/2210.09261.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2022. Ul2: Unifying language learning paradigms.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Sun Tianxiang and Qiu Xipeng. 2023. Moss. *Blog post txsun1997.github.io/blogs/moss.html*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023b. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Ann John, Rishita Anubhai, Smaranda

Muresan, and Dan Roth. 2022. Instruction tuning for few-shot aspect-based sentiment analysis. *ArXiv*, abs/2210.06629.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. *arXiv preprint arXiv:2305.02105*.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022a. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Xiao Wang, Wei Zhou, Can Zu, Han Xia, Tianze Chen, Yuan Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, J. Yang, Siyuan Li, and Chunsai Du. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. *ArXiv*, abs/2304.08085.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hanna Hajishirzi. 2023c. How far can camels go? exploring the state of instruction tuning on open resources. *ArXiv*, abs/2306.04751.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022c. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, M. Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddharth Deepak Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hanna Hajishirzi, and Daniel Khashabi. 2022d. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Conference on Empirical Methods in Natural Language Processing*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022e. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022f. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2021. Reframing human-ai collaboration for generating free-text explanations. *arXiv preprint arXiv:2112.08674*.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir R. Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Conference on Empirical Methods in Natural Language Processing*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023b. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023c. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *ArXiv*, abs/2304.01196.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual nlu. *arXiv preprint arXiv:2004.14353*.

Zhiyang Xu, Ying Shen, and Lifu Huang. 2022. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. *ArXiv*, abs/2212.10773.

Fuzhao Xue, Kabir Jain, Mahir Hitesh Shah, Zangwei Zheng, and Yang You. 2023. Instruction in the wild: A user-based instruction dataset. https://github.com/XueFuzhao/InstructionWild.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*.

Kevin Yang, Nanyun Peng, Yuandong Tian, and Dan Klein. 2022a. Re3: Generating longer stories with recursive reprompting and revision. *arXiv preprint arXiv:2210.06774*.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2022b. Tailor: A prompt-based approach to attribute-based controlled text generation. *ArXiv*, abs/2204.13362.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601.

Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Lu Sheng, Lei Bai, Xiaoshui Huang, Zhiyong Wang, Wanli Ouyang, and Jing Shao. 2023. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *ArXiv*, abs/2306.06687.

YuLan-Chat-Team. 2023. Yulan-chat: An open-source bilingual chatbot. https://github.com/RUC-GSAI/YuLan-Chat.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics.

Ge Zhang, Yemin Shi, Ruibo Liu, Ruibin Yuan, Yizhi Li, Siwei Dong, Yu Shu, Zhaoqun Li, Zekun Wang, Chenghua Lin, Wen-Fen Huang, and Jie Fu. 2023a. Chinese open instruction generalist: A preliminary release. *ArXiv*, abs/2304.07987.

Hang Zhang, Xin Li, and Lidong Bing. 2023b. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. Opt: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023c. Pmc-vqa: Visual instruction tuning for medical visual question answering. *ArXiv*, abs/2305.10415.

Yuanhan Zhang, Qinghong Sun, Yichun Zhou, Zexin He, Zhenfei Yin, Kun Wang, Lu Sheng, Yu Qiao, Jing Shao, and Ziwei Liu. 2022b. Bamboo: Building mega-scale vision dataset continually with human-machine synergy. *arXiv preprint arXiv:2203.07845*.

Yue Zhang, Leyang Cui, Deng Cai, Xinting Huang, Tao Fang, and Wei Bi. 2023d. Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance. *ArXiv*, abs/2305.13225.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, L. Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *ArXiv*, abs/2305.11206.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.