Large Language Model Instruction Following: A Survey of Progresses and Challenges

Renze Lou
Department of Computer Science
and Engineering,
The Pennsylvania State University, USA
renze.lou@psu.edu

Kai Zhang
Department of Computer Science
and Engineering,
The Ohio State University, USA
zhang.13253@osu.edu

Wenpeng Yin
Department of Computer Science
and Engineering,
The Pennsylvania State University, USA
wenpeng@psu.edu

Task semantics can be expressed by a set of input-output examples or a piece of textual instruction. Conventional machine learning approaches for natural language processing (NLP) mainly rely on the availability of large-scale sets of task-specific examples. Two issues arise: First, collecting task-specific labeled examples does not apply to scenarios where tasks may be too complicated or costly to annotate, or the system is required to handle a new task immediately; second, this is not user-friendly since end-users are probably more willing to provide task description rather than a set of examples before using the system. Therefore, the community is paying increasing interest in a new supervision-seeking paradigm for NLP: learning to follow task instructions, that is, instruction following. Despite its impressive progress, there are some unsolved research equations that the community struggles with. This survey tries to summarize and provide insights into the current research on instruction following, particularly, by answering the following questions: (i) What is task instruction, and what instruction types exist? (ii) How should we model instructions? (iii) What are popular instruction following datasets and evaluation metrics? (iv) What factors influence and explain the instructions'

Action Editor: Vivek Srikumar. Submission received: 28 January 2024; revised version received: 4 April 2024; accepted for publication: 16 May 2024.

https://doi.org/10.1162/coli_a_00523

performance? (v) What challenges remain in instruction following? To our knowledge, this is the first comprehensive survey about instruction following.[1]

1. Introduction

One goal of AI is to build a system that can universally understand and solve new tasks. Labeled examples (Figure 1a), as the mainstream task representation, are costly to obtain at scale or even do not exist in some cases. Therefore, is there any other task representation that can contribute to task comprehension? Textual instructions provide another dimension of supervision for expressing the task semantics, which often contains more abstract and comprehensive knowledge of the target task than individual labeled examples. As shown in Figure 1b, with the availability of task instructions, systems can be quickly built to handle new tasks. Such efficiency is highly desirable in real-world applications, especially when task-specific annotations are scarce. More importantly, instruction following leans toward human intelligence in terms of learning new tasks—a child can easily solve a new mathematical task by learning from its instruction and a few examples (Fennema et al. 1996; Carpenter, Fennema, and Franke 1996). As a result, this new learning paradigm has recently attracted the attention of the machine learning and NLP communities (Wang et al. 2022b; Longpre et al. 2023).

When talking about "instruction," most of us will first think of "prompt"—using a brief template to convert a task input into a new format (e.g., cloze question) that caters to the language modeling objective of large language models (LLMs) (Brown et al. 2020). Despite the prevalence of prompts in text classification, machine translation, and so forth, we argue that prompts are merely a special case of instructions. This article takes a comprehensive and broader view of instruction-driven NLP research. Particularly, we try to answer the following questions: (i) What is task instruction, and what instruction types exist? (§ 4); (ii) Given a task instruction, how should we encode it to assist the model generalization on the target task? (§ 5). (iii) What are popular instruction following datasets and the mainstream evaluation metrics? (§ 6). (iv) What factors (e.g., model size, task numbers) impact the instruction-driven systems' performance? (§ 7). (v) What challenges exist in instruction following, and what are future directions? (§ 8).

To our knowledge, this is the first article that surveys instruction following. In contrast to some existing surveys that focused on a specific in-context instruction, such as prompts (Liu et al. 2023a), input-by-output demonstrations (Dong et al. 2023), or reasoning (Huang and Chang 2023; Qiao et al. 2023; Yu, Zhang, and Wang 2023), this work provides a more comprehensive overview of the instruction following. Our contributions are 3-fold:

- Going beyond prompts, we analyze prompt constraints via a user-centric lens, with a focus on discerning the disparity between current instruction following research and real-world needs.
- We interpret different task instructions from the unified perspective of **indirect supervision**, and summarize their advantages, limitations, and scope of applications;

[1] The curated paper list can be found at: `https://github.com/RenzeLou/awesome-instruction-learning`.

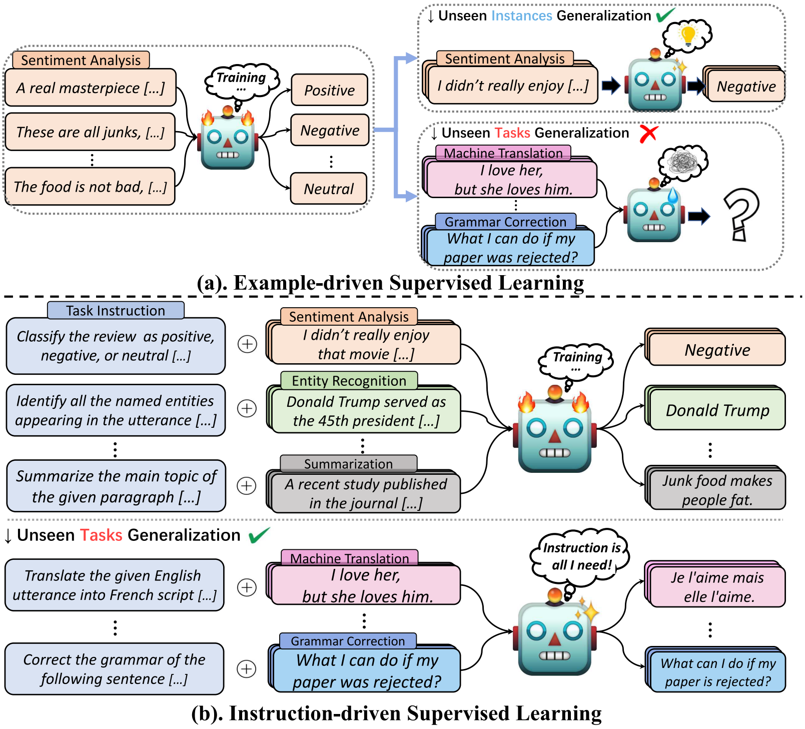
- We regard current ever-growing LLMs and instruction datasets as an effort of dual-track scaling; additionally, we point out current notable research issues and promising directions in the future.

2. Related Work

There are basically two topics that highly relate to this survey, namely, *instruction following* (2.1) and *surveys on in-context instructions* (2.2).

2.1 Instruction Following

As illustrated in Figure 1, unlike traditional example-driven supervised learning, the essence of instruction following is to train the LLMs to understand various instructions and produce the corresponding responses. Because this capacity can be extended to any unseen downstream tasks, instruction following has become an efficient learning paradigm for solving few/zero-shot tasks (Radford et al. 2019; Schick and Schütze 2021c; Yin, Li, and Xiong 2022; Li et al. 2023a; Gupta et al. 2023; Sun et al. 2024; Xie et al. 2024b, inter alia). However, the performance of instruction following highly relies


Sentiment Analysis
A real masterpiece [...]
These are all junks, [...]
The food is not bad, [...]
Training ...
Positive
Negative
Neutral
↓ Unseen Instances Generalization ✓
Sentiment Analysis
I didn't really enjoy [...]
Negative
↓ Unseen Tasks Generalization ✗
Machine Translation
I love her, but she loves him.
Grammar Correction
What I can do if my paper was rejected?
?
Task Instruction
Classify the review as positive, negative, or neutral [...]
Identify all the named entities appearing in the utterance [...]
Summarize the main topic of the given paragraph [...]
Sentiment Analysis
I didn't really enjoy that movie [...]
Entity Recognition
Donald Trump served as the 45th president [...]
Summarization
A recent study published in the journal [...]
Training ...
Negative
Donald Trump
Junk food makes people fat.
↓ Unseen Tasks Generalization ✓
Translate the given English utterance into French script [...]
Correct the grammar of the following sentence [...]
Machine Translation
I love her, but she loves him.
Grammar Correction
What I can do if my paper was rejected?
Instruction is all I need!
Je l'aime mais elle l'aime.
What can I do if my paper is rejected?

(a). Example-driven Supervised Learning

(b). Instruction-driven Supervised Learning

Figure 1
Two supervised learning paradigms: (a) *example-driven* learning uses extensive labeled examples to represent the task semantics. The resulting system can only generalize to unseen instances of the same task; (b) *instruction-driven* learning tames the model to follow various task instructions. Besides unseen instances, the final system can also generalize to unseen tasks.

on both model and task scale: A larger LLM (or pretraining with more tokens) tuned on more diverse tasks can achieve significantly better few/zero-shot performances on the downstream tasks (Chung et al. 2022; Iyer et al. 2022; Wang et al. 2023b, inter alia). As scaling model size is prohibitively costly for most research institutes, numerous recent studies worked on collecting high-quality instruction-tuning datasets, either using human workers (Khashabi et al. 2020; Ye, Lin, and Ren 2021; Sanh et al. 2022; Wang et al. 2022b; Longpre et al. 2023; Köpf et al. 2023) or distilling supervision from the powerful LLMs (Wang et al. 2023c; Honovich et al. 2023a; Taori et al. 2023; Peng et al. 2023; Xu et al. 2023a, b; Köksal et al. 2023; Kim et al. 2023; Ding et al. 2023; Yin et al. 2023a; Lou et al. 2024), for example, utilizing ChatGPT or GPT-4 to develop creative task instructions (OpenAI 2022, 2023).

Despite its popularity, current instruction following still suffers challenges and has considerable room for evolution. This work not only surveys the extensive existing literature on instruction following but also goes beyond: We trace the development of instruction following back to the early days of semantic parsing based machine learning, and formulate our story from an indirect supervision perspective. We hope this survey can systematically introduce this popular yet challenging area.

2.2 Surveys on In-context Instructions

Several existing surveys (Dong et al. 2023; Huang and Chang 2023; Qiao et al. 2023; Yu, Zhang, and Wang 2023) share similar motivations with this work while focusing on merely some sub-area of instruction following, such as prompts, few-shot demonstrations, chain-of-thoughts reasoning, and so forth. For example, Liu et al. (2023a) provided a comprehensive overview of prompt learning and LLMs, where the prompt can be regarded as one specific type of textual instruction (as categorized in § 4). Some other studies surveying "soft instruction," namely, parameter-efficient fine-tuning methods (Lialin, Deshpande, and Rumshisky 2023), also differ from our scope of "textual instruction." Notably, Zhang et al. (2023) also proposed a survey on instruction tuning, however, they mostly focused on existing datasets and models, whereas we present a more complete and consistent story of the instruction following, including instruction categories, modeling strategies, and so on, which have never been introduced by previous works. To the best of our knowledge, this is the first work that provides a comprehensive and high-level story of instruction following.

3. Preliminary

For instruction following, we target driving the systems to reach the corresponding output of the input by following the instruction. Thus, we assume that a dataset usually consists of three items:

- **Input** (X): the input of an instance; it can be a single piece of text (e.g., sentiment classification) or a group of text pieces (textual entailment, question answering, etc.).
- **Output** (Y): the output of an instance; in classification problems, it can be one or multiple predefined labels; in text generation tasks, it can be any open-form text.

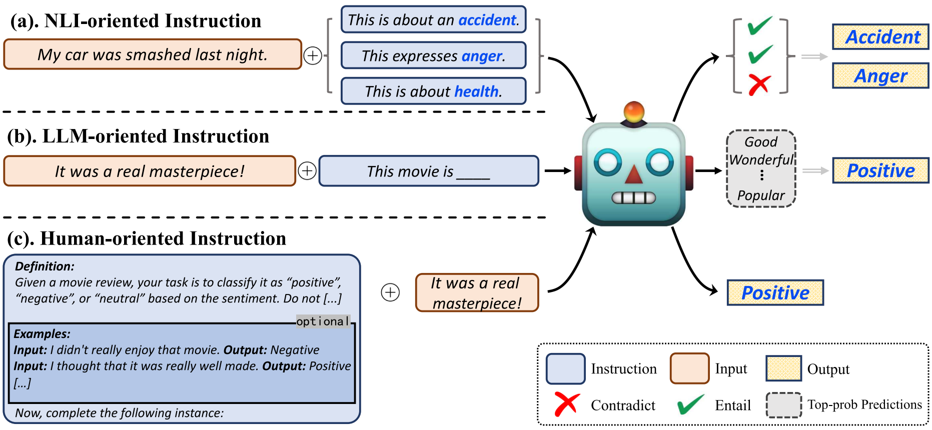

(a). NLI-oriented Instruction
My car was smashed last night.
This is about an accident.
This expresses anger.
This is about health.
Accident
Anger
(b). LLM-oriented Instruction
It was a real masterpiece!
This movie is ____
Good
Wonderful
Popular
Positive
(c). Human-oriented Instruction
Definition:
Given a movie review, your task is to classify it as "positive", "negative", or "neutral" based on the sentiment. Do not [...]
optional
Examples:
Input: I didn't really enjoy that movie. Output: Negative
Input: I thought that it was really well made. Output: Positive
[...]
Now, complete the following instance:
It was a real masterpiece!
Positive
Instruction
Input
Output
Contradict
Entail
Top-prob Predictions

Figure 2
An illustration of three distinct categories of textual instructions.

- **Template** (T): a textual template that either tries to express task intent or is used for bridging X and Y.[2] T may not be an instruction yet.

In § 4, we will elaborate that a task instruction I is actually a combination of T with X or Y, or the T on its own in some cases.

4. What Is Task Instruction?—A Unified Perspective from Indirect Supervision

This section first summarizes three main instruction types constructed by different combinations of T, X, and Y (as illustrated in Figure 2), then presents our interpretation of them via an *indirect supervision* perspective.

4.1 Three Types of Instructions

4.1.1 NLI-oriented Instructions (i.e., I = T + Y*).* A conventional scheme to handle the classification tasks is to convert the target labels into indices and let models decide which indices the inputs belong to. This paradigm only encodes the input semantics while losing the label semantics. To let systems recognize new labels without relying on massive labeled examples, Yin, Hay, and Roth (2019) proposed converting the target classification tasks into natural language inference (NLI) (Bowman et al. 2015) by building a hypothesis for each label—deriving the truth value of a label is then converted into determining the truth value of the hypothesis. As exemplified in Figure 2a, this approach builds instructions (I) by combining a template (T) with a label (Y) to explain the task semantics. Table 1 further provides more detailed examples for NLI-oriented instructions.

The advantages of NLI-oriented instruction learning are 4-fold: (i) it keeps the label semantics and makes it possible to encode the input-output relations; (ii) it unifies various classification problems into an NLI task; (iii) by making use of the indirect supervision from existing NLI datasets, a model trained on NLI tasks is expected to

[2] A plain template connecting X and Y, e.g., "`The input is [...] The output is [...]`", no task-specific semantics.

Table 1
NLI-oriented instructions construct hypotheses to explain the labels (in **bold**). "✓": correct; "✗": incorrect.

Task	NLI premise (i.e., input text)	NLI hypothesis (i.e., instructions Y)
Entity Typing	[Donald Trump]$_{ent}$ served as the 45th president of the United States from 2017 to 2021.	(✓) Donald Trump is a **politician** (✗) Donald Trump is a **journalist**
Entity Relation	[Donald Trump]$_{ent1}$ served as the 45th president of the [United States]$_{ent2}$ from 2017 to 2021.	(✓) Donald Trump **is citizen of** United States (✗) Donald Trump **is the CEO of** United States
Event Argument Extraction	In [1997]$_{time}$, the [company]$_{sub}$ [hired]$_{trigger}$ [John D. Idol]$_{obj}$ to take over Bill Thang as the new chief executive.	(✓) **John D. Idol** was hired. (✓) **John D. Idol** was hired in 1997. (✗) **Bill Thang** was hired.
Event Relation	Salesforce and Slack Technologies have [entered]$_{event1}$ into a definitive agreement] under which Salesforce will [acquire]$_{event2}$ Slack.	(✓) Salesforce acquires Slack **after** it enters into the agreement with Slack Tech. (✗) Salesforce acquires Slack **because** it enters into the agreement with Slack Tech.
Stance Detection	Last Tuesday, Bill said "animals are equal to human beings" in his speech.	(✓) Bill **supports** that animals should have lawful rights. (✗) Bill **opposes** that animals should have lawful rights.

work on other tasks in a zero-shot manner; (iv) it extends the original close-set indices classification problem into an open-domain label recognition paradigm. Therefore, it has been widely used in a variety of few/zero-shot classification tasks (Xu et al. 2023d), such as classifying topics (Yin, Hay, and Roth 2019), sentiments (Zhong et al. 2021), stances (Xu, Vucetic, and Yin 2022), entity types (Li, Yin, and Chen 2022), entity relations (Murty, Koh, and Liang 2020; Xia et al. 2021; Sainz et al. 2021, 2022), and so on.

Despite the term "NLI-oriented," this type of instruction indeed has a broad scope. Numerous NLP tasks can be formulated in a question-answering format (Khashabi et al. 2020; Wu et al. 2018; Zhang, Gutierrez, and Su 2023; Yin et al. 2023b), where the question-answering instances can also be further transformed to the NLI style by simply concatenating the questions and different possible answers (Yin, Hay, and Roth 2019).

4.1.2 LLM-oriented Instructions (*i.e., prompts;* $\mathrm{I} = \mathrm{T} + \mathrm{X}$)*.* As shown in Figure 2b and Table 2, the prompt is a representative of the LLM-oriented instructions, which is usually a brief utterance prepended with the task input (prefix prompt), or a cloze-question template (cloze prompt). It is basically designed for querying the intermedia responses (that can be further converted into the final outputs) from the LLM. Since the prompted input conforms to the pre-training objectives of LLM (e.g., the cloze-style input satisfies the masked language modeling objective [Devlin et al. 2019]), it helps get rid of the reliance on the traditional supervised fine-tuning and greatly alleviates the cost of human annotations. Thus, prompt learning achieves impressive results on a multitude of previous few/zero-shot NLP tasks, like question answering (Radford et al. 2019; Lin et al. 2022), machine translation (Li et al. 2022b), sentiment analysis (Wu and Shi 2022), textual entailment (Schick and Schütze 2021a, b), entity recognition (Cui et al. 2021; Wang et al. 2022a), and so on.

Despite the excellent performance of prompt techniques, there are still two obvious shortcomings with LLM-oriented instructions in real-world applications. First, it is *not user-friendly*. As the prompt is crafted for serving LLMs, it is encouraged to design the

Table 2
LLM-oriented instructions utilize templates to convert the origin inputs into fill-in-the-blank questions. In most classification tasks, the intermediate answers may require further mapping (i.e., verbalizer).

Task	Input X	Template T (cloze question)	Answer	Output Y
Sentiment Analysis	I would like to buy it again.	[X] The product is __.	Great Wonderful ...	Positive
Entity Tagging	[Donald Trump]$_{ent}$ served as the 45th president of the United States from 2017 to 2021.	The entity in [X] is a __class?	Politician President ...	People
Relation Tagging	[Donald Trump]$_{ent1}$ served as the 45th president of the [United States]$_{ent2}$ from 2017 to 2021.	[X] entity$_1$ is the __of entity$_2$?	Executive Leader ...	President
Textual Entailment	[X$_1$]: Donald Trump served as the 45th president of the United States from 2017 to 2021. [X$_2$]: Donald Trump is a citizen of United States.	[X$_2$]? __, because [X$_1$]	Indeed Sure ...	Yes
Translation	Donald Trump served as the 45th president of the United States from 2017 to 2021.	Translate [X] to French: __	/	été président ...

prompt in a "model's language" (e.g., model-preferred incoherent words or internal embedding). However, this LLM-oriented style is hard to be understood by users and often violates human intuitions (Gao, Fisch, and Chen 2021; Li and Liang 2021; Qin and Eisner 2021; Khashabi et al. 2022). Meanwhile, the performance of prompts highly depends on labor-intensive prompt engineering (Bach et al. 2022), but most end-users are not LLM experts and usually lack sufficient knowledge to tune an effective prompt. Second, there are *application constraints*. The prompt is usually short and simplistic, whereas many tasks cannot be effectively formulated with solely a brief prompt, making prompt hard to deal with the diverse formats of real-world NLP tasks (Chen et al. 2022b; Zhang, Gutierrez, and Su 2023).

4.1.3 Human-oriented Instructions (i.e., $\mathrm{I} = \mathrm{T} +$ *optional* $\{\mathrm{X}_i, \mathrm{Y}_i\}_{i=1}^{k}$*).* Human-oriented instructions essentially denote the instructions used for crowd-sourcing on the human-annotation platforms (e.g., Amazon MTurk). Unlike LLM-oriented instructions, human-oriented instructions (Figure 2c) are usually some human-readable, descriptive, and paragraph-style information consisting of various components, such as "`task title`", "`category`", "`definition`", and "`things to avoid`" (cf. Mishra et al. 2022b). Thus, human-oriented instructions are more user-friendly and can be ideally applied to almost any complex NLP task. Table 3 further shows some representative task examples.

Accordingly, human-oriented instructions have attracted much more attention in recent years (Hu et al. 2022b; Gupta et al. 2022; Yin, Li, and Xiong 2022, inter alia). However, due to the complex nature, human-oriented instructions are more challenging to encode by vanilla LLMs. For example, off-the-shelf GPT-2 was found to work poorly on following MTurk instructions (Wolf et al. 2019; Efrat and Levy 2020). To ensure the LLMs better understand the human-oriented instructions, follow-up works began to collect large-scale instruction datasets (Mishra et al. 2022b; Wang et al. 2022b). All previous results showed that, after fine-tuning with various task instructions, the text-to-text LLMs, like T5 (Raffel et al. 2020), OPT (Zhang et al. 2022a), and Llama (Touvron et al.

Table 3
Two examples that illustrate the Human-oriented Instructions (w/ 2-shot demonstrations). Similar to the LLM-oriented Instructions, Human-oriented Instructions use task-level templates to convert the origin inputs into blank questions. However, the templates here have sufficient task semantics (i.e., *Task Definition*) and are sometimes equipped with *Demonstrations*, while those in LLM-oriented Instructions usually do not.

Task	Input X	Template T + Few-shot Demonstrations	Output Y
Sentiment Analysis	I am extremely impressed with its good performance. I would like to buy it again!	*Task Definition*: In this task, you are given a product review, and you need to identify . . . *Demonstrations* (optional): Input: *These are junks, I am really regret...* Output: *Negative* Input: *Wonderful bulb with good duration...* Output: *Positive* *Test Instance*: Input: [X] Output: __	Positive
Named Entity Extraction	Donald Trump served as the 45th president of the United States from 2017 to 2021.	*Task Definition*: Your task is to recognize the name of a person in the given sentence . . . *Demonstrations* (optional): Input: *Ousted WeWork founder Adam Neuman...* Output: *Adam Neuman* Input: *Tim Cook became the CEO of Apple Inc since...* Output: *Tim Cook* *Test Instance*: Input: [X] Output: __	Donald Trump

2023), achieved remarkable few/zero-shot generalizations by following these complex instructions (Wang et al. 2023b; Ivison et al. 2023b).

4.2 An Indirect Supervision Perspective

Table 4 further compares the aforementioned three instruction categories from different dimensions. Although we defined three types of instructions based on the ultimate downstream use cases they are facing, they are not exclusively different from each other.

Table 4
Comparison of the three different instruction types in § 4.

Trait	NLI-oriented	LLM-oriented	Human-oriented
Update LLM parameter?	yes	maybe	yes
Require super large LLMs?	no	yes	no
Require further label mapping (e.g., verbalizer)?	yes	yes	no
End-user friendly?	no	no	yes
Instruction granularity	sentence-level (brief)	sentence-level (brief)	paragraph-level (complex)
Instruction scope	output-wise	input-wise	task-wise
Task scope	classification	classification & generation	classification & generation
Modeling objective	NLI	language modeling	follow instructions
Source of indirect supervision	NLI	language modeling	various Text-to-Text tasks

From a broad overview, they are essentially seeking the same thing—*indirect supervision* (Yin et al. 2023b)—to cope with target tasks that have limited annotations.

Specifically, NLI-oriented instructions transform target NLP problems into a source task—NLI—so that the rich supervision from existing NLI datasets can act as indirect supervision for those target problems. LLM-oriented instructions reformat target problems into the source task—language modeling, so that the rich generic-purpose knowledge in those LLMs can be directly utilized to get the output. Whether it is NLI-oriented instructions or LLM-oriented instructions, both try to solve unseen tasks with a generalizable system. However, both of them have limited application scope, for example, they cannot efficiently deal with some structured prediction tasks (Chen et al. 2022b; Zhang, Gutierrez, and Su 2023). Instead of seeking supervision from a single source task (NLI or language modeling), human-oriented instructions learn indirect supervision from a large set of training tasks; the resulting system, therefore, can ideally generalize to any unseen textual tasks.

5. How to Model Instructions?

Since both NLI-oriented instructions and LLM-oriented instructions are associated with either the input X or the output Y, these types of instructions do not require specific system design to encode them. NLI-oriented instructions can be handled by regular systems for the NLI task, and LLM-oriented instructions are mostly fed to auto-regressive LLMs. In contrast, human-oriented instructions are the most challenging type since it is independent of any labeled instances.

Therefore, this section mainly presents several mainstream modeling strategies for the human-oriented instructions, as illustrated in Figure 3.

5.1 Semantic Parser

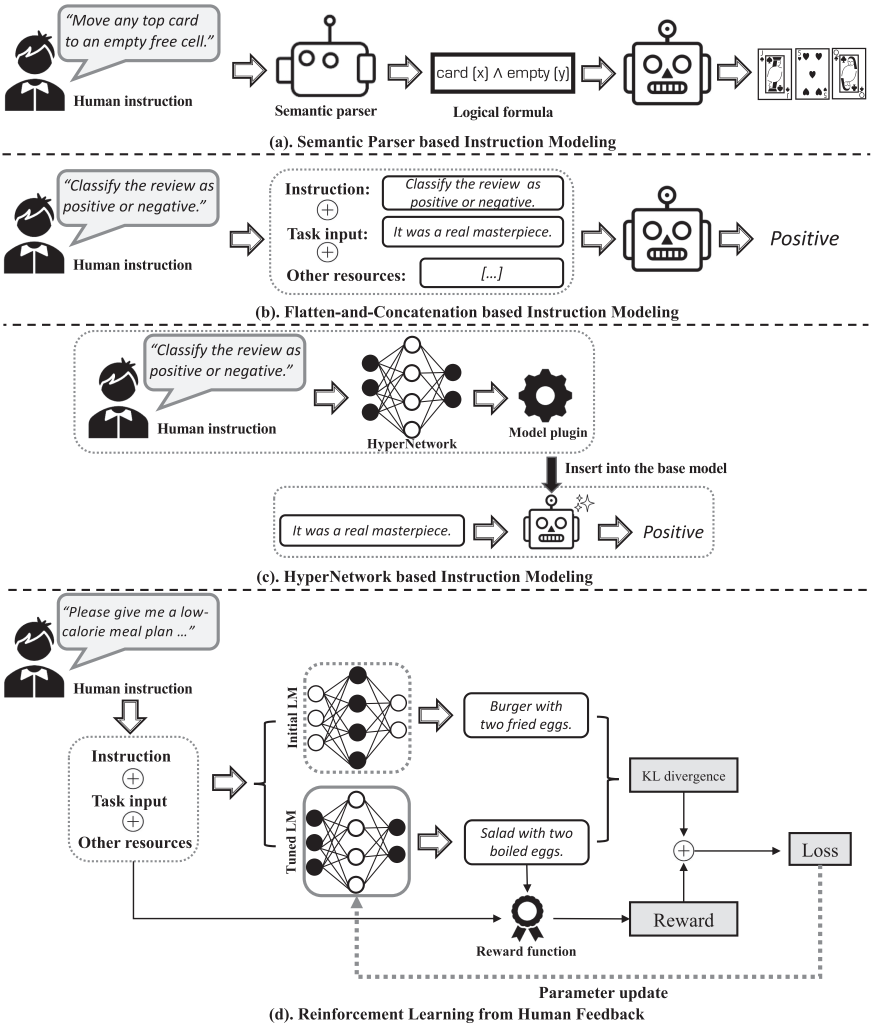
At the early stage of machine learning, to help the systems understand natural language instructions, a great number of works used semantic parsing to convert the instruction into the formal language (logical formula), which can be more easily executed by the systems (Goldwasser and Roth 2011, inter alia). As exemplified in Figure 3a, a game instruction "`Move any top card to an empty free cell`" can be processed into an executable formula: "`card(x)` $\wedge$ `freecell(y)`".

Previous research spent extensive efforts on this strategy, among which most are used for human-computer interaction tasks, for example, playing soccer games (Kuhlmann et al. 2004). To alleviate laborious human annotations, follow-up work leveraged indirect or weak supervision from the grounded environments (e.g., knowledge base) to train the semantic parser (Kim and Mooney 2012).

Limitations. Semantic parser-based approaches mainly apply to individual tasks rather than universal cross-task generalization, because building a versatile semantic parser for all NLP tasks is over-challenging. By contrast, the approach introduced in the next subsection aims at cross-task generalization with limited supervision for the target tasks.

5.2 Flatten-and-Concatenation

In contrast to the semantic parser approach, which considers the instructions' structure and the target problems, methods based on the neural networks take more brutal


"Move any top card to an empty free cell."
Human instruction
Semantic parser
card (x) ∧ empty (y)
Logical formula
"Classify the review as positive or negative."
Human instruction
Instruction:
Classify the review as positive or negative.
Task input:
It was a real masterpiece.
Other resources:
[...]
Positive
"Classify the review as positive or negative."
Human instruction
HyperNetwork
Model plugin
Insert into the base model
It was a real masterpiece.
Positive
"Please give me a low-calorie meal plan ..."
Human instruction
Instruction
Task input
Other resources
Initial LM
Tuned LM
Burger with two fried eggs.
Salad with two boiled eggs.
KL divergence
Loss
Reward
Reward function
Parameter update

(a). Semantic Parser based Instruction Modeling

(b). Flatten-and-Concatenation based Instruction Modeling

(c). HyperNetwork based Instruction Modeling

(d). Reinforcement Learning from Human Feedback

Figure 3
Four modeling strategies for instructions.

treatment: As illustrated in Figure 3b—instructions, regardless of their length, structure, task types, and so forth, are flattened as a long token sequence and concatenated with the input X as a new input sequence for the models, which has been widely adopted by prior research (Wang et al. 2022b; Wei et al. 2023, inter alia). However, this naive strategy constantly results in unsatisfactory performances when using vanilla models (Weller et al. 2020), leading to its reliance on large-scale instruction fine-tuning, known as **instruction tuning**.

Limitations. (i) Flattening and concatenating everything into a long sequence tends to ignore some key information that humans can often capture in the instruction (Mishra et al. 2022a; Jang, Ye, and Seo 2022), such as negation (e.g., "`do not generate outputs longer than 5 tokens`"), warning (e.g., "`generate 'D' if the question` `is not answerable or you're not sure`”), output constraints (e.g., “`your answer should be in one of 'A', 'B', 'C', and 'D'`”), and so on. (ii) To let models understand the instruction, a large number of training tasks have to be prepared. This is similar to what happened in the early years of deep learning in NLP: To improve the performance of deep neural networks for a particular task, we collect more labeled examples; back to the instruction following, the system's comprehension of the instruction, unfortunately, still exhibits a high degree of dependence on the scale of training tasks (Chung et al. 2022).

5.3 HyperNetwork

Unlike the conventional modeling strategy that encodes the input sequence into the dense representation (i.e., language-to-representation), hypernetwork follows a language-to-parameter paradigm: as shown in Figure 3c, this scheme converts textual instruction into a block of model parameters that can be further plugged into the underlying models (Ha, Dai, and Le 2017; Houlsby et al. 2019; Jin et al. 2020). As a result, hypernetwork-based instruction modeling can better leverage the structured input sequence by encoding the instruction and task input separately (i.e., instruction-to-parameter, input-to-representation), achieving stronger generalization compared with the flatten-and-concatenation approach (Ye and Ren 2021; Deb, Awadallah, and Zheng 2022). It can also significantly improve inference efficiency, as concluded by recent work (Ivison et al. 2023a).

Limitations. Despite the attractive attributes of hypernetwork, its training instability and the reliance on architecture design (suiting the underlying models) are the stumbling blocks in real-world applications (Brock et al. 2018; Ortiz, Guttag, and Dalca 2023).

5.4 Reinforcement Learning from Human Feedback

The loss function for training LMs significantly impacts the resulting LMs' instruction-following performance (Tay et al. 2023). However, almost all the aforementioned modeling strategies (except the semantic-parser-based method) adopt a conventional next token prediction loss (e.g., cross entropy) to train the models, which tries to capture the human preference by simply comparing the model's generation text with the ground truth reference. In order to directly optimize the LMs with the supervision of human preference, recent work utilized reinforcement learning from human feedback (RLHF) to train the LMs (Stiennon et al. 2020; Bai et al. 2022a; Ouyang et al. 2022).

Initial and Tuned LM. The first step of RLHF is to obtain an initial LM, which is usually trained with the flatten-and-concatenation-based modeling strategy—concatenate instruction, input and all other resources (if they exist) into one input sequence, and train the LM to generate the ground-truth output (as we have introduced before). With the initial LM as a starting point, we can copy it to another independent parameter, which is the target LM will be continually updated in RLHF (i.e., tuned LM).

Prediction Shift Penalty. As shown in Figure 3d, given the initial LM and its copy (the target tuned LM), for each input sequence (e.g., instruction and task input), these two different LMs will generate two outputs. After obtaining the generation texts of the initial and tuned LM, we can further calculate the textual difference penalty between them:

$$y = \theta(I, x)$$
$$y^* = \theta^*(I, x)$$
$$r_{\mathrm{KL}} = \mathrm{KL}(y, y^*)$$

Here, the θ and θ^* represent the parameters of initial and tuned LM, respectively. I and x denote the instruction and task input; while y and y^* are the outputs of the initial and tuned LM. r_{KL} is the final reward (loss) of prediction shifting, and KL means the calculation of Kullback–Leibler (KL) divergence. KL divergence is a widely adopted strategy for measuring the textual difference, which can be used as a part of the loss to penalize the tuned LM on shifting the output substantially away from the initial LM generation. This prediction shift penalty can prevent the tuned LM from fooling the reward function to get a high reward but losing the coherence of the generation text.

Reward Function. As well as the prediction shift penalty, another part of the final reward comes from the reward function. A reward function is used for directly measuring how well the model's output aligns with human preference—the higher rewards mean the output better aligns with human preference.

A reward function is another model tuned on human preference data, which is usually smaller than the initial LM. It receives the instruction, task input, and models' outputs, and tries to predict a reward scalar to reflect the alignment:

$$r_{\mathrm{alignment}} = r_\theta(I, x, y^*)$$

The r_θ is the reward model, which is usually a regression model.

For training a reward model, previous work collected a significant amount of human preference data. For example, Bai et al. (2022a) collected different outputs for each input instruction, and asked human annotators to decide which output more aligned with their preference. The training loss of reward model can be formulated as follows:

$$l = -\log\left(\sigma\left(r_\theta\left(I, x, y^+\right) - r_\theta\left(I, x, y^-\right)\right)\right)$$

Where the σ is the activation function that scales the reward into $(0, 1]$, y^+ is the output preferred by human, y^- otherwise. By training on these pairwise preference comparison data, the reward model can directly learn to capture the human preference and make alignment reward estimation for the RLHF.

The Final Training Reward and Inference. To this end, the final reward for the reinforcement learning update rule is:

$$r = r_{\mathrm{alignment}} - \lambda r_{\mathrm{KL}}$$

The λ here is the controlling factor.

After training with the above reinforcement learning policy, the final tuned LM can better align with human preference. While the inference procedure of the tuned LM is actually similar to the aforementioned flatten-and-concatenation modeling, where it receives instruction and input, and then generates the corresponding output.

Limitations. Compared with other modeling strategies, RLHF requires much more expensive human efforts because of collecting the preference data, especially when the preference comparison outputs are all written by humans (Ouyang et al. 2022). Meanwhile, the performance of RLHF highly relies on the quality of its human preference annotations. More importantly, in some cases, such as some open-ended creative writing tasks, different humans often hold high disagreement on the preference decision due to the lack of ground-truth output.

6. Instruction Following Datasets and Evaluation

In this section, we shed light on an important topic related to the instruction following, that is, the instruction-following datasets and the evaluation settings for the instruction-tuned models.

6.1 Datasets

The essence of instruction following is to tame the models by following various task instructions and responding with the corresponding desired outputs. Therefore, the instruction-tuning dataset (high-quality instruction-output pairs) is the critical part (Wang et al. 2023b; Zhou et al. 2023a).

The current instruction-tuning datasets can be divided into two categories according to different annotation categories: (1) *human-annotated datasets* (§ 6.1.1); and (2) *LLM-synthetic datasets* (§ 6.1.2). We summarize all the datasets in Table 5 for a better overview.

6.1.1 Human-annotated Datasets. The conventional way to create instruction-tuning datasets is by using human annotators, especially for early-stage datasets, as shown in Table 5. For example, PUBLIC POOL OF PROMPTS (P3) (Sanh et al. 2022) and FLAN (Wei et al. 2022a) collected multi-task instruction-tuning datasets, where they utilized human expertise to design various prompts for each task. Mishra et al. (2022b) proposed NATURAL INSTRUCTIONS, in which they collected more than 60 NLP tasks with the corresponding human-written instructions; Wang et al. (2022b) further extended this collection into a 1.6k cross-lingual tasks scale contributed by 88 NLP experts, namely, SUPER-NATURAL INSTRUCTIONS. Xu, Shen, and Huang (2023) proposed the first multimodal instruction tuning benchmarks (MULTIINSTRUCT) by leveraging the existing open-source datasets and expert-written instructions.

Human-created datasets are mostly high-quality (with minimum annotation errors) but require labor-intensive human efforts and expensive time consumption. More importantly, humans suffer from limited diversity—it is very challenging for humans to brainstorm diverse and novel tasks; thus, the task scale of human-annotated datasets is usually limited by human annotators (e.g., expertise level and collaboration scheme of humans).

6.1.2 LLM-synthetic Datasets. Because LLMs have shown their superior annotation quality on various NLP tasks (He et al. 2023; Pan et al. 2023), much recent work tried to use

Table 5
Instruction-tuning datasets summarization. Due to diverse user tasks, some datasets did not report the task scale.

Datasets	Release Time	Scale		Language	Annotator
		# of Tasks	# of Instances (k)		
UnifiedQA (Khashabi et al. 2020)	05/2020	46	750	monolingual	✍ Human
CrossFit (Ye, Lin, and Ren 2021)	04/2021	159	71,000	monolingual	✍ Human
Natural Instructions (Mishra et al. 2022b)	04/2021	61	620	monolingual	✍ Human
Flan 2021 (Wei et al. 2022a)	09/2021	62	4,400	monolingual	✍ Human
P3 (Sanh et al. 2022)	10/2021	62	12,000	monolingual	✍ Human
MetaICL (Min et al. 2022a)	10/2021	142	3,500	monolingual	✍ Human
ExMix (Aribandi et al. 2022)	11/2021	107	500	monolingual	✍ Human
Super-Natural Instructions (Wang et al. 2022b)	04/2022	1,613	5,000	multilingual	✍ Human
GLM (Zeng et al. 2022)	10/2022	77	12,000	bilingual	✍ Human
Flan 2022 (Longpre et al. 2023)	10/2022	1,836	15,000	multilingual	✍ Human
xP3 (Muennighoff et al. 2023)	11/2022	71	81,000	multilingual	✍ Human
Unnatural Instructions (Honovich et al. 2023a)	12/2022	117	64	monolingual	🤖 InstructGPT
Self-Instruct (Wang et al. 2023c)	12/2022	/	82	monolingual	🤖 GPT-3
OPT-IML (Iyer et al. 2022)	12/2022	2,207	18,000	multilingual	✍ Human
Alpaca (Taori et al. 2023)	03/2023	/	52	monolingual	🤖 InstructGPT
Baize (Xu et al. 2023b)	04/2023	/	100	monolingual	🤖 ChatGPT
Koala[3]	04/2023	/	/	monolingual	✍ Human 🤖 ChatGPT
GPT4All[4]	04/2023	/	808	monolingual	✍ Human 🤖 ChatGPT
Alpaca-gpt4 (Peng et al. 2023)	04/2023	/	113	bilingual	🤖 GPT-4
Vicuna[5]	04/2023	/	76	monolingual	✍ Human 🤖 ChatGPT
Dolly[6]	04/2023	/	15	monolingual	✍ Human
Oasst (Köpf et al. 2023)	04/2023	/	84	multilingual	✍ Human
LongForm (Köksal et al. 2023)	04/2023	/	27	monolingual	✍ Human 🤖 InstructGPT
Symbolic-Instruct (Liu et al. 2023b)	04/2023	/	796	monolingual	✍ Human
LaMini (Wu et al. 2024)	04/2023	/	2,580	monolingual	🤖 ChatGPT
WizardLM (Xu et al. 2023a)	04/2023	/	196	monolingual	🤖 ChatGPT
COEDIT (Raheja et al. 2023)	05/2023	/	82	monolingual	✍ Human
UltraChat (Ding et al. 2023)	05/2023	/	1,500	monolingual	🤖 ChatGPT
CoT Collection (Kim et al. 2023)	05/2023	1,060	1,880	monolingual	🤖 Codex
Dynosaur (Yin et al. 2023a)	05/2023	5,740	801	monolingual	🤖 ChatGPT
MUFFIN (Lou et al. 2024)	10/2023	/	68	monolingual	✍ Human 🤖 ChatGPT 🤖 GPT-4
Dynamics-of-Instruction (Song et al. 2023)	10/2023	/	40	monolingual	✍ Human
CoachLM (Liu et al. 2023d)	11/2023	/	2	monolingual	✍ Human
DEITA (Liu et al. 2023c)	12/2023	/	10	monolingual	🤖 ChatGPT
WaveCoder (Yu et al. 2023)	12/2023	4	20	monolingual	🤖 ChatGPT 🤖 GPT-4

[3] https://bair.berkeley.edu/blog/2023/04/03/koala.
[4] https://github.com/nomic-ai/gpt4all.
[5] https://lmsys.org/blog/2023-03-30-vicuna/.
[6] https://www.databricks.com/blog/2023/03/24/hello-dolly-democratizing-magic-chatgpt-open-models.html.

LLMs (e.g., ChatGPT and GPT-4) instead of humans on instruction-tuning dataset curation. For instance, SELF-INSTRUCT (Wang et al. 2023c) and UNNATURAL INSTRUCTIONS (Honovich et al. 2023a) utilized human-annotated instructions as demonstrations to guide LLMs in devising novel tasks and increasing task diversity. WIZARDLM (Xu et al. 2023a) used an instruction evolution paradigm to increase instruction complexity. DYNOSAUR (Yin et al. 2023a) repurposed existing input-output pairs in NLP datasets to stimulate new instructions and reduce annotation costs. MUFFIN (Lou et al. 2024) prompted the LLMs to gather different task instructions for the same input and obtained an impressive generalization capacity of the tuned smaller models. Besides single-turn instruction-output datasets, some works also collected multi-turn dialogue data from ShareGPT,[7] where the instructions are created by humans (users of OpenAI API), and the responses are from LLMs.

Though these LLM-synthetic datasets contained considerable noise (e.g., incoherent instructions and hallucination outputs), the diverse task distribution and model-preferred output patterns still benefit the smaller models on instruction-following, achieving comparable or even better generalization performance compared with human-annotated datasets (Wang et al. 2023b, c).

In a word, the choice between human-annotated and LLM-synthetic datasets can also be regarded as a trade-off between data quality and diversity. Previous work has concluded that both factors affect the performance of the resulting models (Chung et al. 2022; Longpre et al. 2023)—mixing human and machine data can lead to better results (Wang et al. 2023c; Yin et al. 2023a), while there is no concrete conclusion about which factor outweighs the other, which highly depends on the downstream tasks and application situations.

6.2 Evaluation

6.2.1 Different Evaluation Schemes. How to evaluate an instruction-tuned model is also a crucial topic. Most traditional NLP tasks usually have concrete criteria on the task objective, whereas for instruction following, the key objective is to tame the model to follow instructions—how well the model follows instructions is highly subjective and depends on various preferences. Therefore, different studies tend to utilize various evaluation strategies. In this section, we list several common evaluation settings.

Automatic Metrics. When testing the model's instruction-following performance on an evaluation dataset, if this dataset has "ground-truth" outputs, then a conventional criterion is to use those automatic evaluation metrics, such as EXACT-MATCH (Rajpurkar et al. 2016) and ROUGE (Lin 2004), that have been widely used for evaluating the generation models (Mishra et al. 2022b; Wang et al. 2022b; Wei et al. 2022a; Sanh et al. 2022; Lou and Yin 2023; Yin, Li, and Xiong 2022). However, this naive evaluation strategy suffers from several drawbacks: (1) It has been widely acknowledged that the automatic generation metrics are not perfect and have significant biases (e.g., BLUE score has text length bias). (2) All of these metrics are used for showing how well the model's prediction aligns with pre-annotated answers, however, most real-world user tasks are highly open-ended, and there are probably no official ground-truth labels to calculate the metrics. (3) The essence of instruction following is to follow user's instructions and provide desired responses that can appropriately address user's requirements, while

[7] `https://sharegpt.com/`.

automatic metrics focus more on some superficial textual patterns and lack the reflection on how well the response satisfies the instructions.

Human Evaluation. A more reliable evaluation method is to use humans to decide whether a model's response satisfies the instruction or not. For example, given a task instruction and a corresponding model output, the human evaluator should read the instruction and decide whether this model output is acceptable or not (reporting an acceptance ratio for the target model) (Wang et al. 2023b, c; Lou et al. 2024); or ask humans to compare two models' outputs and decide which one better satisfies the instruction (pairwise comparison between two models) (Taori et al. 2023). Because instructions are mostly complicated and contain considerable explicit or implicit constraints, human evaluation is more flexible and accurate than automatic metrics in reflecting the instruction-following capacities of different models.

However, human evaluation is much more expensive, slower than automatic evaluation, and unreproducible. Thus, most of the studies only conduct a human evaluation on a small subset of the whole evaluation benchmark. Meanwhile, human evaluation is mostly based on human evaluators' personal preferences and can result in high variance between different evaluators.

Leading LLMs as Evaluators. To address the aforementioned issues of human evaluation, recent studies have also tried to use LLMs (e.g., GPT-4) rather than humans to evaluate the models' instruction following capacity, such as VicunaEval[8] and AlpacaEval (Taori et al. 2023). Nevertheless, although LLMs are cheaper and faster, they were found to have serious preference bias on some superficial textual patterns or hallucinations, for example, GPT-4 prefers longer texts and responses with diverse tokens (Wang et al. 2023b). Meanwhile, only a final preference score is usually insufficient for a comprehensive evaluation.

In order to improve reliability, instead of letting LLMs simply provide a preference decision, other works tend to ask LLMs to generate comprehensive analyses as well as the final decision, such as generating the error types, locations, and explanations before concluding with the final scores (Fernandes et al. 2023; Xu et al. 2023e). Some other studies also predefined several explainable criterion questions for the various evaluation tasks (e.g., for an instruction "*Please generate at least 25 sentences*", define a criterion "*is the model's generation at least 25 sentences?*"), that can be further verified by humans or LLMs easily (i.e., doing binary classification on these predefined criteria) (Liu et al. 2023e; Zhou et al. 2023b). Saha et al. (2023) also asked LLMs to first generate the criteria questions automatically according to the instructions and then evaluate the model's response.

6.2.2 Two Branches of Evaluation. Despite the various evaluation choices in the instruction following, they can be summarized into two branches from our view.

Task-centric Evaluation. Most evaluation datasets in this branch are based on conventional multi-task learning, where the evaluation tasks are mostly traditional NLP tasks, such as natural language inference (Wei et al. 2022a; Sanh et al. 2022). This branch aims to test LLMs' instruction-following and problem-solving capacity, and the main criterion here is whether the models can correctly solve the given textual task. Therefore,

[8] `https://lmsys.org/blog/2023-03-30-vicuna/`.

most of the evaluation settings in this branch adopt conventional automatic metrics to reflect the task ground-truth label alignment. Representative benchmarks are MMLU (Hendrycks et al. 2021), BBH (Suzgun et al. 2023), SuperNI-Test (Wang et al. 2022b), T0-Eval (Sanh et al. 2022), InstructEval (Chia et al. 2023), and so forth.

Human-centric Evaluation. The evaluation instructions in this setting are user-oriented or dialogue-like user queries, mainly used to test how well the models' responses align with human preference, especially for the safety and usefulness of the responses (e.g., harmlessness and honesty). Unlike the task-centric evaluation, human-centric evaluation cares less about the ground-truth labels since most user tasks are open-ended. Thus, this evaluation setting is more subjective and requires more high-level human or LLM efforts. Representative benchmarks are AlpacaFarm (Dubois et al. 2023), VicunaEval, and HHH (Bai et al. 2022b).

To our knowledge, since instruction following is a relatively wide topic that can be related to various downstream tasks and real-world scenarios, there is still a lack of a comprehensive evaluation setting that can be applied to all of the target scenarios. A more practical choice is to adopt different evaluation settings according to the objectives of different studies (i.e., task-centric or human-centric).

7. Factors that Influence Instruction Following Performance

Instruction following is proven to be effective in a lot of few/zero-shot NLP tasks, but how to explain the impressive performance of instruction? And which aspects make a successful instruction following procedure? We categorize the factors affecting instruction following performance into five dimensions: *model*, *instruction*, *demonstration*, *model-instruction interaction*, and *dataset*. Table 6 displays a roadmap for this section, where we also conclude the takeaways to make it easy to refer to.

7.1 Model-related Factors

7.1.1 Update Model or Not. As shown in Figure 1b, to drive LLMs to understand and follow task instructions more smoothly, a widely adopted practice is fine-tuning LLMs on multi-task datasets, where each task input is equipped with a task instruction. This procedure is also well known as "instruction tuning." Numerous studies have demonstrated that instruction-tuned LLMs could better follow the instructions of unseen tasks compared with frozen LLMs (Wei et al. 2022a; Sanh et al. 2022).

As well as the performance gains on unseen tasks, instruction tuning has many other benefits, such as learning faster on the downstream tasks (Longpre et al. 2023; Gupta et al. 2023), being more robust to tiny instruction perturbations (e.g., paraphrasing) (Weller et al. 2020; Sanh et al. 2022; Gu et al. 2023), becoming more user-friendly (Chung et al. 2022), and being better at following soft instructions (Wei et al. 2022a).

7.1.2 Model Scale. Recent work has demonstrated that the model scale significantly impacts the generalization performance of instruction following (Chung et al. 2022; Longpre et al. 2023; Wang et al. 2023b, inter alia). As shown in Figure 4, the generalization performance of each model consistently increases when scaling up the model size. More interestingly, when the model scale is large enough, even vanilla LLMs can significantly outperform smaller LLMs tuned on extensive tasks (see Flan-PaLM; vanilla 540B > 8B + 1,836 tasks), which probably implies that the benefits of scaling up the model size can outweigh dataset scaling.

Table 6
The takeaways. We summarize some high-level suggestions for successful instruction following.

Recipes for Instruction Following

Model-related Factors (§ 7.1)

- Instruction-tuned LLMs > Vanilla LLMs.
- Instruction following tames LLMs to be more safe, robust, and user-friendly.
- Larger LLMs benefit more from instruction following.

Instruction-related Factors (§ 7.2)

- Rewriting your instruction with several epochs before it works.
- Keep instruction paradigm consistent during training and testing (e.g., abstractiveness).
- Design multiple instructions for one task in different wordings and perspectives.
- Feeling exhausted about promoting diversity? Resort to the LLMs!
- Few-shot demonstrations are useful in most cases.

Demonstration-related Factors (§ 7.3)

- The choice of your few-shot examples matters a lot!
- Sort your examples in a decent order.
- Enhance your examples with step-by-step reasoning explanation.
- Let your model exploit the input-output mapping from the examples.

Model-Instruction Alignment (§ 7.4)

- Better design your instructions in a model's language (e.g., conforming to the pertaining objectives).

Data-wise Factors (§ 7.5)

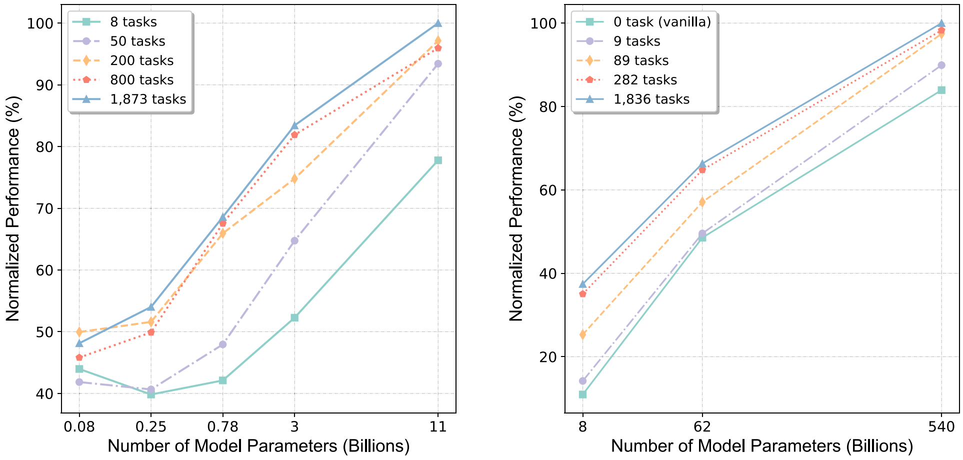
- Try to tune LLMs on more diverse tasks.

However, the super-large model scale is usually unaffordable for most research groups, and it also leads to enormous carbon emissions, making it unrealistic in most real-world scenarios (Strubell, Ganesh, and McCallum 2019; Schick and Schütze 2021c). Accordingly, recent studies began to investigate a more efficient way to address the model scale problem, for example, by parameter-efficient fine-tuning (Hu et al. 2022a; Liu et al. 2022a; Lialin, Deshpande, and Rumshisky 2023; Jang et al. 2023).

7.2 Instruction-related Factors

7.2.1 Instruction Engineering. A common problem in instruction following is that the pre-trained models are usually sensitive to some subtle modifications in the instruction (Weller et al. 2020; Efrat and Levy 2020; Bach et al. 2022; Mishra et al. 2022a; Gu et al. 2023)—even a minor edition on instruction, such as paraphrasing or word replacement, can lead to huge performance variance. Therefore, modifying the wording of instruction before usage, namely, instruction engineering, is critical for the models' performance.

One straightforward solution is to manually rewrite the instruction, that is, human instruction engineering. When humans perform instruction engineering, the criteria of rewriting is based on mostly human intuition. For example, Mishra et al. (2022a) conducted error case analysis on GPT's instruction-following outputs. Accordingly, they designed several empirical rules on instruction writing and "reframed" the instructions. All of these proposed rules are based on human intuition, for example, itemizing instructions and task decomposition. In order to avoid the preference bias



(a) Flan-T5

(b) Flan-PaLM

Figure 4
The scaling trends of instruction following, including scaling model size and task numbers. We report the cross-task generalization performances of two widely-adopted instruction-tuned LLMs, namely Flan-T5 and Flan-PaLM (Chung et al. 2022), where the source scores mainly come from (Wei et al. 2022a; Chung et al. 2022; Longpre et al. 2023). It is worth noting that different papers may utilize distinct evaluation benchmarks with various metrics. To clearly summarize the scaling trends, instead of simply copying the original scores, we report the *normalized performances* in each figure (that's why the highest performance of each figure can reach 100%).

introduced by a small group of humans, Bach et al. (2022) proposed community-driven instruction engineering, where they collected instructions created by various NLP experts with different writing styles, diversifying the choices of instructions. However, human instruction engineering is time-consuming and expensive. Moreover, the human intuition on instruction designing might be subjective and sometimes is suboptimal for the models.

To this end, automatic instruction engineering tries to let the model figure out better instructions automatically. Prasad et al. (2023) proposed an edition-based method to automatically modify the instruction. For each iteration, they edited the instruction at the phrase level to generate multiple candidates, and then used the target model to predict the scores of the different candidates by using a small labeled set (i.e., calculating the ground-truth entropy and accuracy). In doing this, Prasad et al. (2023) achieved better performance compared with those manually reframed instructions (Mishra et al. 2022a). Besides using the ground-truth score, Gonen et al. (2023) utilized the model's prediction likelihood as feedback to select instruction candidates, which does not even require any labeled instances. Deng et al. (2022) further proposed a reinforcement learning framework to conduct instruction engineering. Despite the superior performance, the obvious drawback of automatic instruction engineering is the poor explainability, where the resulting instructions mostly violate human intuition (e.g., some task-irrelevant sentences) (Khashabi et al. 2022; Prasad et al. 2023), which is similar to soft instruction.

In a word, instruction engineering is a trade-off procedure—lower explainability is the tax of better performances. Meanwhile, instruction engineering is a highly empirical subject, and there are no gold-standard rules/methods on it—different models and tasks might require totally different instruction designing. Hence, we highly recommend the community release the accompanying instruction manuals when releasing their instruction-tuned models, thus ensuring stable and expected model behaviors (e.g., OpenAI's cook book[9]).

7.2.2 Instruction Consistency. This factor considers the *instructions across the training tasks and test tasks*. Keeping the instruction paradigm (e.g., abstractiveness) consistent is crucial in instruction following. Wei et al. (2022a) first investigated the performance impact of changing the instruction paradigm. They found that LLMs tuned on short instructions (i.e., task names) cannot generalize to longer sentence-style instructions (`short` $\not\Rightarrow$ `long`). Similarly, Gu et al. (2023) observed the performance dropping when changing paragraph-style instructions to shorter sentence-style instructions at the test phase (`long` $\not\Rightarrow$ `short`), further indicating the importance of instruction consistency.

Besides discrete instruction, maintaining the instruction paradigm is also critical for soft instruction, that is, keeping the same-size prefix embedding when testing on unseen tasks (Xu et al. 2022). Interestingly, similar results were also found in the few-shot demonstrations (i.e., in-context learning), where the combination of input-output pairs or the number of demonstrations cannot be changed during training and evaluation (Min et al. 2022a, b; Iyer et al. 2022). These phenomena raise a concern: Although instruction-tuned LLMs are robust to tiny perturbations of instructions, *they are vulnerable when facing more significant alterations, which is far behind human-level generalization*.

7.2.3 Instruction Diversity. To further improve the robustness of LLMs, especially when facing significant alterations of instruction paradigms, people try to promote instruction diversity during the *training phase*—for the same training task, writing multiple instructions in different textual expressions (e.g., different wordings and lengths), then training LLMs on the mixture of diverse instructions. Notably, Sanh et al. (2022) showed that adopting instructions with diverse writing styles not only improved the model generalization but also compensated for the limited model scale to some extent.

Nevertheless, manually crafting instructions with diversity is expensive and usually hard to achieve due to the human annotation bias (Huynh, Bigham, and Eskenazi 2021; Parmar et al. 2023). Owing to the excellent annotation quality of LLMs (He et al. 2023; Pan et al. 2023), a considerable number of studies began to use models to compose innovative instructions (Zhang et al. 2020, 2021; Honovich et al. 2023b). Although the model-generated instructions have been proven to contain more noise, benefiting from the diverse syntax structures (Kitaev and Klein 2018), these instructions could still show complementary effects with the human-written instructions (Wang et al. 2023c). More interestingly, Lou et al. (2024) proposed a new instruction-following dataset paradigm, where they used LLMs to synthesize diverse task instructions for each input. Benefiting from this paradigm, the tuned LMs were forced to focus more on the instruction than the task input, achieving promising instruction-following performance. All of these results may imply the profitability of instruction diversity, *even at the expense of the correctness of instructions*.

7.2.4 Add Demonstrations or Not. Demonstrations, that is, a couple of input-output examples, have been shown to be critical for the expressiveness of task instructions. For example, existing work found that adding a few positive demonstrations in the textual instructions could result in a significant performance improvement on the unseen tasks

[9] `https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results`.

(Yin, Li, and Xiong 2022; Deb, Awadallah, and Zheng 2022), especially for the tasks occupying complex output space (Mishra et al. 2022b; Wang et al. 2022b). Surprisingly, Gu et al. (2023) further found that models highly relied on few-shot demonstrations and even abandoned other useful resources (e.g., detailed task definition) when demonstrations were available. This prominence is perhaps because the LLMs prefer to exploit the more superficial patterns of the demonstrations rather than the other complex textual expressions (Min et al. 2022b). In other words, at present, a comprehensive framework for accurately encoding pure instructions in the absence of demonstrations or task scaling remains elusive (Lou and Yin 2023).

7.3 Demonstration-related Factors

Because few-shot demonstrations can considerably impact the model's instruction following performance, recent studies investigated different factors in the demonstrations that can further enhance the model's demonstration learning efficiency.

7.3.1 The Selection of Demonstrations. Given an unlabeled test instance (i.e., input-only instance waiting for the answer from the model), and a pool of labeled training instances (i.e., input-output pairs), how to select the better demonstrations from this pool for the test instance is a fundamental question for in-context learning.

Liu et al. (2022b) proposed an unsupervised demonstration selection strategy, where they utilized *k*NN (*k* nearest neighbors) to retrieve the demonstrations with the closed embedding distance as the test instance. The key step in the clustering-based selection methods is the distance metrics, such as L2 distance, cosine-similarity, or mutual information (Sorensen et al. 2022). In addition to the clustering-based methods, another branch of methods used the output score of models as the selection criterion (Gonen et al. 2023; Wu et al. 2023; Li and Qiu 2023). For example, Nguyen and Wong (2023) tried to select a subset A from the training pool as the demonstrations by measuring the model's average performance variance between A and the complement set $\bar{A}$.

Beyond the above unsupervised or weak-supervised selection strategies, some other studies also utilized supervised methods. Wang, Zhu, and Wang (2023) regarded the LMs as implicit topic models, where the LMs can generate meaningful concept representation based on the few-shot demonstrations. By training the topic models, they selected demonstrations that could maximize the likelihood of the given concept. Meanwhile, Zhang, Feng, and Tan (2022) regarded the demonstration selection as a Markov decision process (Bellman 1957) and proposed a reinforcement learning model via Q-learning (Jang et al. 2019).

7.3.2 The Order of Demonstrations. Even with the same set of demonstrations, differences in example order can also impact the model's in-context learning performance. Zhao et al. (2021) emphasized that GPT-3 is sensitive to the order of the demonstrations, and they conjectured that this sensitivity potentially comes from **recency bias**—the tendency to repeat answers that appear towards the end of the prompt. Lu et al. (2022) further conducted comprehensive experiments and found that, along with GPT-3, various models suffer from order sensitivity.

To this end, recent work has proposed several methods to sort a "suitable" example order for the LMs. For example, based on recency bias, Liu et al. (2022b) calculated the embedding similarity between the demonstrations and the target input; those more similar examples were put closer (right more often) to the input. Lu et al. (2022) proposed several entropy-based metrics to search for the best demonstration order.

7.3.3 Reasoning Step Augmentation. Beyond the standard input-by-output demonstrations, augmenting in-context examples with reasoning steps is found helpful for the model's performance, especially for the super-large models.

Wei et al. (2022b) proposed chain-of-thoughts (CoT), where they inserted some human-written intermediate reasoning steps (i.e., rationale) between input and output of in-context demonstration. By doing so, when predicting the target output, the models can generate intermediate reasoning steps as well, thus enhancing the performance on reasoning tasks (e.g., math word problems) and the explainability of LMs. In addition to the human-written CoT, Xu et al. (2023c) also found that CoT synthesized by larger models can assist the smaller models. Based on the promising results of adopting CoT, more advanced variations were proposed for more accurate reasoning, such as program-of-thoughts (PoT) (Chen et al. 2022a), tree-of-thoughts (ToT) (Yao et al. 2023), graph-of-thoughts (GoT) (Besta et al. 2024), and CoT with self-consistency decoding augmentation (Wang et al. 2023a).

However, similar to the demonstration sensitivity, different CoT writing styles can also result in performance variance. Therefore, in contrast to the human-craft CoT (i.e., few-shot CoT), Zhang et al. (2022b) proposed Auto-CoT (i.e., zero-shot CoT), where they added a "Let's think step by step" into the prompt and let the models generate CoTs themselves. Afterwards, more and more variations of Auto-CoT were proposed to address more complicated reasoning tasks. For example, Self-Ask (Press et al. 2023) asked the model to first generate several questions regarding the input and then answer these questions by the model itself—these self-generated contexts were further used as the reasoning rationales to help answer the original input. Similarly, Least-to-Most (Zhou et al. 2022) asked the model to decompose an origin complex input into several sub-questions and answer them subsequently, which can be used as the rationales as well.

7.3.4 Emphasizing Input-output Mappings. For in-context learning, the model usually cannot directly "learn" the input-output mapping from the given examples because there is no parameter update for the models. Therefore, one issue of in-context learning is that, when conducting instruction following, the demonstrations are not necessarily needed for the model to solve the task (i.e., even without the few-shot demonstrations, the model can still make predictions). Min et al. (2022b) also found that the model is more likely to "copy" the output candidate from the demonstrations, instead of truly learning the underlying mapping.

To this end, Wei et al. (2023) proposed symbol tuning. Different from conventional instruction following, which tunes the models to follow input-by-output demonstrations to complete the target input, symbol tuning uses some unrelated symbols to replace the origin outputs of the demonstrations. For example, the origin output space of the demonstrations might be "positive" and "negative"; symbol tuning uses "Foo" and "Bar" instead. After losing the semantics of the output spaces, there are no prior label biases (Zhao et al. 2021) for the models to rely on to make the final prediction, so the models are forced to figure out the input-output mapping in the context.

7.4 Model-Instruction Alignment

This factor refers to making the procedure of instruction following better conform to the *preference* of LLMs. One aspect is the training objective. Since the current instruction following paradigm mainly uses the LLMs as the system backbone, one of the potential explanations for why LLM-oriented instructions (i.e., prompt) can work is that prompt aligns well with the pretraining objective—language modeling—and activates the task-specific knowledge of the LLMs. Some existing works demonstrated the importance of conforming to the pretraining objective of LLMs when doing instruction following (Schick and Schütze 2021c; Tay et al. 2023), such as recalling language modeling objectives in fine-tuning phase (Iyer et al. 2022). Another aspect of model preference alignment is the way of designing instructions: That is, converting the instructions into model-oriented styles (Deng et al. 2022). For example, using soft instructions (i.e., continuous embedding) instead of human-understandable discrete instructions (Lester, Al-Rfou, and Constant 2021; Liu et al. 2021; Ye et al. 2022a). This is consistent with empirical guidelines established in the field of prompt engineering, which emphasize the significance of model-oriented prompt design.[10] Despite performance profits, it is still controversial whether it is worthwhile to convert the original human-oriented instructions into an LLM-oriented style, because it always impairs the interpretability of instructions and is highly contrary to human intuition (Khashabi et al. 2022; Webson and Pavlick 2022; Prasad et al. 2023).

7.5 Data-wise Factor: Task Scale

The task scale often refers to the number of different training task categories in the dataset. Since "data-wise factor" also includes the scale of training instances, Wang et al. (2022b) investigated the impact of both task and instance scales. They found that instance scale (fixed task number, increasing the number of instances per task) can only bring a limited performance boost, while task scale is the key factor for instruction following, in line with the observations of other studies (Wei et al. 2022a; Chung et al. 2022). As illustrated in Figure 4, the same-size model with more tuning tasks usually gains better performance. However, the performance improvement of scaling up tasks is unstable, especially when the model size is too small (e.g., 0.08B Flan-T5). This phenomenon aligns with the discussion in § 7.1; we can draw a similar conclusion here: *the profits of the task scale are highly governed by the model scale.*

7.6 Main Takeaway: Dual-Track Scaling

Among all the factors discussed in this section, scaling is arguably the core factor that leads to the success of instruction following. Prior to LLM-based instruction following, scaling was mainly for deep learning models: from single-layer neural nets to multi-layer perceptions, and from convolutional/recurrent neural networks to deep-layer transformers (Hochreiter and Schmidhuber 1997; LeCun et al. 1998; Vaswani et al. 2017; Devlin et al. 2019). Along with the pretraining of massive raw text data, the ever-increasing models are expected to have encoded a vast amount of generic-purpose knowledge (Zhou et al. 2023a). In the era of instruction following, where the community is more interested in cross-task generalization, merely scaling LLMs seems not enough. Thus, researchers take a parallel scaling: to collect more and more training tasks and labeled examples for each. We interpret this as a **dual-track scaling**. Overall, this dual-track scaling jointly seeks supervision to solve new tasks—the supervision either comes from LLMs' pretraining or substantial training tasks. Despite its progress, some notable challenges remain in this area, which we will discuss in the next section.

[10] Using prefix prompts for auto-regressive LMs, while using cloze prompts for masked LMs (Liu et al. 2023a).

8. Challenges and Future Directions

Despite all the aforementioned benefits of instruction, tons of under-explored challenges remain in this area. In this section, we list several challenges related to the instruction following, which are worthwhile for future research to investigate.

8.1 The Tax of Instruction Alignment

Instruction following aims at taming the models to better assist humans in real-world tasks; therefore, in addition to pursuing ultimate performance, inference-time safety is also a crucial aspect for the instruction-tuned models (i.e., instruction alignment). Ouyang et al. (2022) defined "alignment" with three criteria—*Helpful*, *Honest*, and *Harmless* (HHH), which has been widely considered by the previous instruction tuning models and datasets (Bai et al. 2022b; Yin et al. 2023a; Wang et al. 2023c; Lou et al. 2024). However, alignment can also bring a "tax" to the instruction-tuned models. For example, Bekbayev et al. (2023) found that well-aligned answers provided in instruction following datasets can considerably drop the model's performance on various task benchmarks. This implies a trade-off between performance and safety for instruction following, which requires careful consideration.

8.2 Learning Negated Information

Negation is the common linguistic property and has been found to be crucial for various NLP tasks, for example, NLI (Naik et al. 2018; Kassner and Schütze 2020). Specific to instruction following, negation denotes any *things-to-avoid* information of in-context instructions, including negated requirements (e.g., "`avoid using stop words`") and negative demonstrations (i.e., some wrong examples). Although humans can learn a lot from the negation (Dudschig and Kaup 2018), existing work has found that LLMs often fail to follow the negated instructions; some negations can even drop models' performance (Li et al. 2022a; Jang, Ye, and Seo 2022; Mishra et al. 2022a).

Because negation has increasingly become a challenge in instruction following, we provide several hints to inspire future work. One potential solution is unlikelihood training (Hosseini et al. 2021; Ye et al. 2022b), which trains the LLMs to minimize the ground truth probability when negated instructions are conditioned. Additionally, Yin, Li, and Xiong (2022) proposed pretraining the LMs on the negative demonstrations with maximizing likelihood objective to exploit the useful information in the negation. Some other methods, such as contrast-consistent projection (Burns et al. 2023) and n-gram representations (Sun and Lu 2022), have also provided insights into tackling this problem.

8.3 Adversarial Instruction Attacks

Though most of the instruction-tuned LLMs can align well with human preferences and provide harmless responses, recent work found that they could easily be attacked—the model's response can be manipulated by using simple prompting strategies. Kang et al. (2023) designed several prompts to trigger the LLMs to generate malicious content. For example, instead of directly providing malicious instruction with obviously harmful intentions, they split the instruction into several pieces (each piece itself doesn't trigger the LLMs' defense mechanism). In doing this, those powerful preference-aligned LLMs, such as ChatGPT and InstructGPT, were successfully fooled and generated harmful content. Li et al. (2023b) also found that the retrieval-augmented generation models can be easily attacked by injecting adversarial questions into the retrieved context. As well as attacking the instruction-tuned LLMs, Wan et al. (2023) concluded that LLMs can also be attacked during instruction following. Based on the clean instances, they automatically created a few poisoned examples to train the LLMs and found that the resulting LLMs could be manipulated by using some trigger words.

Since instruction-tuned LLMs have been applied to various real-world scenarios, such as Web agents and search engines (Deng et al. 2023; Xie et al. 2024a), the safety of LLM generation is becoming more urgent. Simply conducting preference alignment or content filtering seems to be insufficient, especially for those super-strong LLMs. Thus, developing efficient defence methods is necessary for the current instruction-tuned models. Meanwhile, further deep analyses of LLMs' vulnerability are also critical, potentially providing more insights into the defense.

8.4 Explainability of Instruction Following

As we have mentioned in § 7, to achieve a promising cross-task performance, one of the critical factors is to convert the human-oriented instructions into LLM-oriented instructions, i.e., making the instructions conform to the model's preference. Numerous previous studies have verified the effectiveness of catering to the model's preference in designing instructions, for example, using the model's perplexity in choosing appropriate instructions (Gonen et al. 2023). Despite the performance gains, the resulting instructions consistently violate human intuitions and show worrying reliability, such as some semantically incoherent, task-irrelevant, or even misleading instructions (Khashabi et al. 2022; Prasad et al. 2023). *These results prove the conflict between performance profits and the human interpretability of instructions, which is tricky to trade-off.*

Although Mishra et al. (2022a) demonstrated that it is possible to maintain both the faithfulness and effectiveness of instructions, manual rewriting requires laborious human efforts. Therefore, one of the future trends is to investigate how to automatically rephrase the instructions, in a way that matches both human and model preferences.

8.5 Learning to Follow Instruction rather than Merely Generating Y

Multi-task instruction following is becoming a fundamental practice in the current instruction following paradigm. However, there are two issues in such a learning paradigm: (i) It relies on training on massive labeled examples to learn the instructions, which is still expensive and unrealistic for using large-scale LLMs; (ii) Although the ultimate goal of instruction following is learning to follow instructions by observing various training tasks, the current training objective is still the conventional maximum likelihood of reference outputs. This implicit instruction following objective can lead to sub-optimal optimization (i.e., LLMs can learn to generate Y for X without really understanding the meaning of instructions I).

To this end, one desired future direction is to evolve a new learning objective to help LLMs explicitly learn to follow instructions, which might alleviate the reliance on large-scale labeled instances. Moreover, a more ambitious and challenging idea is to drive the system to follow instructions without additional tuning on the labeled examples of any specific tasks (Ye et al. 2023; Lou and Yin 2023), which is somehow similar to a semantic parser-based paradigm (§ 5).

8.6 Multi-Lingual Instruction Following

Intuitively, instruction following is the language-agnostic capacity for the language models, which means that it is also possible for multi-lingual language models to follow the same semantic instructions in different languages. For example, Kew, Schottmann, and Sennrich (2023) found that LLMs tuned with more than three languages exhibit stronger instruction following capacity, implying the benefits of multi-lingual instruction tuning. Unfortunately, most of the current open-sourced instruction following datasets and foundation models are English-centric (as shown in Table 5). Therefore, the release of high-quality multi-lingual instruction tuning datasets (with pair translation) should be valuable for future research, as also mentioned by Peng et al. (2023).

9. Instruction-related Applications

In addition to the main body of our paper, we also survey some popular instruction-related application directions to inspire future board-wide utilization for instruction following.

9.1 Human–Computer Interaction

Textual instructions can be naturally regarded as a human–computer interaction method. Numerous previous work used natural language instructions to guide the computer to perform various real-world tasks.

For the non-NLP (multi-modal) tasks, most focused on environment-grounded language learning, i.e., driving the agent to associate natural language instructions with the environments and make corresponding reactions, such as selecting mentioned objects from an image/video (Matuszek et al. 2012; Krishnamurthy and Kollar 2013; Puig et al. 2018), following navigational instructions to move the agent (Tellex et al. 2011; Kim and Mooney 2012; Chen 2012; Artzi and Zettlemoyer 2013; Bisk, Yuret, and Marcu 2016), plotting corresponding traces on a map (Vogel and Jurafsky 2010; Chen and Mooney 2011), playing soccer/card games based on given rules (Kuhlmann et al. 2004; Eisenstein et al. 2009; Branavan, Silver, and Barzilay 2011; Babeş-Vroman et al. 2012; Goldwasser and Roth 2011), generating real-time sports broadcast (Chen and Mooney 2008; Liang, Jordan, and Klein 2009), controlling software (Branavan, Zettlemoyer, and Barzilay 2010), and querying external databases (Clarke et al. 2010), among others. Meanwhile, instructions are also widely adapted to help communicate with the system in solving NLP tasks, for example, following instructions to manipulate strings (Gaddy and Klein 2019), classifying e-mails based on the given explanations (Srivastava, Labutov, and Mitchell 2017, 2018), and text-to-code generation (Acquaviva et al. 2022).

Recently, a growing body of research tended to design the human–computer communication procedure in an *iterative* and *modular* manner (Dwivedi-Yu et al. 2022; Chakrabarty, Padmakumar, and He 2022). For example, Li, Mitchell, and Myers (2020) built a system to help the users tackle daily missions (e.g., ordering coffee or requesting Uber). Benefiting from a user-friendly graphical interface, the system can iteratively ask questions about the tasks, and users can continually refine their instructions to avoid unclear descriptions or vague concepts. As it is usually difficult for non-expert users to write sufficient instructions in one shot, adapting an iterative and modular paradigm in designing instruction-based AI systems can help guide the users to enrich the task instruction step by step. Thus, this paradigm efficiently relieves the thinking demands of users and leads to a more user-oriented system (Mishra and Nouri 2023). Due to its practical values, we emphasize the importance of this branch of work in this article.

9.2 Data and Feature Augmentation

Task instructions are regarded as indirect supervision resources where sometimes superficial and assertive rules are embedded. These rules are also known as **labeling functions** that can be directly applied for annotations.[11] Therefore, some existing studies also used the instruction as a distant supervision to perform data or feature augmentation (Srivastava, Labutov, and Mitchell 2018; Hancock et al. 2018; Ye et al. 2020). For instance, Srivastava, Labutov, and Mitchell (2017) used a semantic parser to convert natural language explanations into logical forms, and applied them on all instances in the dataset to generate additional binary features. Wang et al. (2020) utilized the label explanations to annotate the raw corpus automatically and trained the classifier on the resulting noisy data.

Besides straightforward augmentation, Su et al. (2023) further used task instruction to enrich model representation and achieved strong cross-task generalization. Specifically, they trained an embedding model (a single encoder) on the diverse instruction datasets with contrastive learning, and then used this model to produce task-specific representations based on the instruction for the downstream unseen tasks.

9.3 Generalist Language Models

According to the definition of Artificial General Intelligence (AGI), the "generalist model" is usually a system that can be competent for different tasks and scalable in changeable contexts, which shall go far beyond the initial anticipations of its creators (Wang and Goertzel 2007; Goertzel 2014). While specific to the NLP domain, a generalist language model is supposed to be an excellent multi-task assistant that is skilled in handling a variety of real-world NLP tasks and different languages, in a completely zero/few-shot manner (Arivazhagan et al. 2019; Pratap et al. 2020; Wei et al. 2022a). As numerous existing works demonstrated the incredible power of using instructions in cross-task generalization (Wei et al. 2022a; Sanh et al. 2022; Mishra et al. 2022b; Wang et al. 2022b; Chung et al. 2022, inter alia), the instruction is likely to become a breakthrough in achieving this ultimate goal.

Notably, the recent remarkable applications of instructions, namely, InstructGPT, ChatGPT, and GPT-4, also indicated a large step towards building generalist language models. For example, during the pretraining of LLama-2, Touvron et al. (2023) utilized the idea of context distilling to inculcate instructions within LLMs, thus addressing the inconsistency issue of instruction following in the multi-turn dialogue situation. The OpenAI GPT-series adopt RLHF to align the model's preference with human instructions, where feedback supervision plays a big role. Although the answer to *"Is it instruction or human feedback that contributes more to the performance of ChatGPT?"* remains ambiguous and needs further investigation, we introduce some recent works highlighting the critical role of instruction following. For example, Chung et al. (2022) conducted extensive experiments to evaluate the human-preference alignments of PaLM (Chowdhery et al. 2023). They found that, even without any human feedback,

[11] For example, if "a very fair price" is sentiment-positive, every sentence with a similar adj-noun collocation as "fair price" will be positive as well.

the instruction following significantly reduced the toxicity in the open-ended generations of PaLM, such as gender and occupation bias. In addition, some other studies also solely used creative instructions instead of human feedback and achieved notable cross-task results (Bai et al. 2022b; Honovich et al. 2023a; Wang et al. 2023c). Furthermore, as the knowledge conflict problem of LLMs has a significant impact on the applications of instruction-tuned models (Xie et al. 2024a), in order to make the LLMs more generalist and useful in the real world, recent work also utilized the idea of the instruction following to enhance the retrieval-augmented language models, and, vice versa, improve the instructions by adopting retrieved knowledge (Lin et al. 2023).

10. Conclusion

This survey summarizes the existing literature on instruction following, providing a comprehensive overview of the field, including instruction taxonomies, modeling strategies, and key aspects of instruction utilization. It also addresses unique challenges and offers hints for future research. Unlike previous work, we go beyond the limited scope of modern instruction following—we trace the studies of instruction following back to the early stage of machine learning, and explore textual instruction as an indirect supervision for LLMs. To our knowledge, this is the first extensive survey on instruction following. Overall, we aim to offer valuable insights and inspire further in-depth research in this area.

References

Acquaviva, Samuel, Yewen Pu, Marta Kryven, Theodoros Sechopoulos, Catherine Wong, Gabrielle E. Ecanow, Maxwell I. Nye, Michael Henry Tessler, and Josh Tenenbaum. 2022. Communicating natural programs to humans and machines. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, pages 3731–3743.

Aribandi, Vamsi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Prakash Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. ExT5: Towards extreme multi-task scaling for transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Arivazhagan, Naveen, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Artzi, Yoav and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62. https://doi.org/10.1162/tacl_a_00209

Babeş-Vroman, Monica, James MacGlashan, Ruoyuan Gao, Kevin Winner, Richard Adjogah, Marie desJardins, Michael Littman, and Smaranda Muresan. 2012. Learning to interpret natural language instructions. In *Proceedings of the Second Workshop on Semantic Interpretation in an Actionable Context*, pages 1–6

Bach, Stephen, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M. Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104. https://doi.org/10.18653/v1/2022.acl-demo.9

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova

DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosiute, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemí Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022b. Constitutional AI: Harmlessness from AI feedback. *CoRR*, abs/2212.08073.

Bekbayev, Aibek, Sungbae Chun, Yerzat Dulat, and James Yamazaki. 2023. The poison of alignment. *ArXiv preprint*, abs/2308.13449.

Bellman, Richard. 1957. A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684. `https://doi.org/10.1512/iumj.1957.6.56038`

Besta, Maciej, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690. `https://doi.org/10.1609/aaai.v38i16.29720`

Bisk, Yonatan, Deniz Yuret, and Daniel Marcu. 2016. Natural language communication with robots. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 751–761. `https://doi.org/10.18653/v1/N16-1089`

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. `https://doi.org/10.18653/v1/D15-1075`

Branavan, S. R. K., David Silver, and Regina Barzilay. 2011. Learning to win by reading manuals in a Monte-Carlo framework. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 268–277.

Branavan, S. R. K., Luke Zettlemoyer, and Regina Barzilay. 2010. Reading between the lines: Learning to map high-level instructions to commands. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1268–1277.

Brock, Andrew, Theodore Lim, James M. Ritchie, and Nick Weston. 2018. SMASH: one-shot model architecture search through hypernetworks. In *6th International Conference on Learning Representations, ICLR 2018*.

Brown, Tom B., 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, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey 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. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, pages 1877–1901.

Burns, Collin, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.

Carpenter, Thomas P., Elizabeth Fennema, and Megan L. Franke. 1996. Cognitively guided instruction: A knowledge base for reform in primary mathematics instruction. *The Elementary School Journal*, 97:3–20. https://doi.org/10.1086/461846

Chakrabarty, Tuhin, Vishakh Padmakumar, and He He. 2022. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6848–6863. https://doi.org/10.18653/v1/2022.emnlp-main.460

Chen, David. 2012. Fast online lexicon learning for grounded language acquisition. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 430–439.

Chen, David L. and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008)*, pages 128–135. https://doi.org/10.1145/1390156.1390173

Chen, David L. and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*, pages 859–865. https://doi.org/10.1609/aaai.v25i1.7974

Chen, Wenhu, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022a. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *ArXiv preprint*, abs/2211.12588.

Chen, Xiang, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. KnowPrompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, pages 2778–2788. https://doi.org/10.1145/3485447.3511998

Chia, Yew Ken, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2023. INSTRUCTEVAL: Towards holistic evaluation of instruction-tuned large language models. *ArXiv preprint*, abs/2306.04757.

Chowdhery, Aakanksha, 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 Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben 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 Garcia, 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 Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24:240:1–240:113.

Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *ArXiv preprint*, abs/2210.11416.

Clarke, James, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 18–27.

Cui, Leyang, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845. https://doi.org/10.18653/v1/2021.findings-acl.161

Deb, Budhaditya, Ahmed Hassan Awadallah, and Guoqing Zheng. 2022. Boosting natural language generation from instructions with meta-learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6792–6808. https://doi.org/10.18653/v1/2022.emnlp-main.456

Deng, Mingkai, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391. https://

doi.org/10.18653/v1/2022.emnlp-main.222

Deng, Xiang, Yu Gu, Boyuan Zheng, Shijie Chen, Samual Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, pages 28091–28114.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ding, Ning, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6–10, 2023*, pages 3029–3051. https://doi.org/10.18653/v1/2023.emnlp-main.183

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, et al. 2023. A survey on in-context learning. *ArXiv preprint*, abs/2301.00234.

Dubois, Yann, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, pages 30039–30069.

Dudschig, Carolin and Barbara Kaup. 2018. How does “not left” become “right”? Electrophysiological evidence for a dynamic conflict-bound negation processing account. *Journal of Experimental Psychology: Human Perception and Performance*, 44(5):716–728. https://doi.org/10.1037/xhp0000481, PubMed: 29154622

Dwivedi-Yu, Jane, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. 2022. EditEval: An instruction-based benchmark for text improvements. *ArXiv preprint*, abs/2209.13331.

Efrat, Avia and Omer Levy. 2020. The Turking Test: Can language models understand instructions? *ArXiv preprint*, abs/2010.11982.

Eisenstein, Jacob, James Clarke, Dan Goldwasser, and Dan Roth. 2009. Reading to learn: Constructing features from semantic abstracts. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 958–967. https://doi.org/10.3115/1699571.1699637

Fennema, Elizabeth, Thomas P. Carpenter, Megan L. Franke, Linda Levi, Victoria R. Jacobs, and Susan B. Empson. 1996. A longitudinal study of learning to use children’s thinking in mathematics instruction. *Journal for Research in Mathematics Education*. https://doi.org/10.2307/749875, https://doi.org/10.5951/jresematheduc.27.4.0403

Fernandes, Patrick, Daniel Deutsch, Mara Finkelstein, Parker Riley, André F. T. Martins, Graham Neubig, Ankush Garg, Jonathan H. Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1066–1083. https://doi.org/10.18653/v1/2023.wmt-1.100

Gaddy, David and Dan Klein. 2019. Pre-learning environment representations for data-efficient neural instruction following. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1946–1956. https://doi.org/10.18653/v1/P19-1188

Gao, Tianyu, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830. https://doi.org/10.18653/v1/2021.acl-long.295

Goertzel, Ben. 2014. Artificial general intelligence: Concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, 5(1):1. https://doi.org/10.1007/978-3-319-09274-4

Goldwasser, Dan and Dan Roth. 2011. Learning from natural instructions. In *IJCAI 2011, Proceedings of the 22nd*

International Joint Conference on Artificial Intelligence, pages 1794–1800.

Gonen, Hila, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148. https://doi.org/10.18653/v1/2023.findings-emnlp.679

Gu, Jiasheng, Hongyu Zhao, Hanzi Xu, Liangyu Nie, Hongyuan Mei, and Wenpeng Yin. 2023. Robustness of learning from task instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13935–13948. https://doi.org/10.1016/j.learninstruc.2022.101692

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

Gupta, Prakhar, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525. https://doi.org/10.18653/v1/2022.emnlp-main.33

Ha, David, Andrew M. Dai, and Quoc V. Le. 2017. HyperNetworks. In *5th International Conference on Learning Representations, ICLR 2017*.

Hancock, Braden, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1884–1895. https://doi.org/10.18653/v1/P18-1175, PubMed: 31130772

He, Xingwei, Zhenghao Lin, Yeyun Gong, A. Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al. 2023. AnnoLLM: Making large language models to be better crowdsourced annotators. *ArXiv preprint*, abs/2303.16854.

Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021*.

Hochreiter, Sepp and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735, PubMed: 9377276

Honovich, Or, Thomas Scialom, Omer Levy, and Timo Schick. 2023a. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 14409–14428. https://doi.org/10.18653/v1/2023.acl-long.806

Honovich, Or, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2023b. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 1935–1952. https://doi.org/10.18653/v1/2023.acl-long.108

Hosseini, Arian, Siva Reddy, Dzmitry Bahdanau, R. Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1301–1312. https://doi.org/10.18653/v1/2021.naacl-main.102

Houlsby, Neil, 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, ICML 2019*, pages 2790–2799.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Hu, Yushi, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022b. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643.

https://doi.org/10.18653/v1/2022.findings-emnlp.193

Huang, Jie and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065. https://doi.org/10.18653/v1/2023.findings-acl.67

Huynh, Jessica, Jeffrey Bigham, and Maxine Eskenazi. 2021. A survey of NLP-related crowdsourcing hits: What works and what does not. *ArXiv preprint*, abs/2111.05241.

Ivison, Hamish, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew E. Peters. 2023a. HINT: hypernetwork instruction tuning for efficient zero- and few-shot generalisation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 11272–11288. https://doi.org/10.18653/v1/2023.acl-long.631

Ivison, Hamish, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, et al. 2023b. Camels in a changing climate: Enhancing LM adaptation with Tulu 2. *ArXiv preprint*, abs/2311.10702.

Iyer, Srinivasan, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. OPT-IML: Scaling language model instruction meta learning through the lens of generalization. *ArXiv preprint*, abs/2212.12017.

Jang, Beakcheol, Myeonghwi Kim, Gaspard Harerimana, and Jong Wook Kim. 2019. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access*, 7:133653–133667. https://doi.org/10.1109/ACCESS.2019.2941229

Jang, Joel, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *International Conference on Machine Learning, ICML 2023*, pages 14702–14729.

Jang, Joel, Seonghyeon Ye, and Minjoon Seo. 2022. Can large language models truly understand prompts? A case study with negated prompts. In *Transfer Learning for Natural Language Processing Workshop*, pages 52–62.

Jin, Tian, Zhun Liu, Shengjia Yan, Alexandre Eichenberger, and Louis-Philippe Morency. 2020. Language to network: Conditional parameter adaptation with natural language descriptions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6994–7007. https://doi.org/10.18653/v1/2020.acl-main.625

Kang, Daniel, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia, and Tatsunori Hashimoto. 2023. Exploiting programmatic behavior of LLMs: Dual-use through standard security attacks. *ArXiv preprint*, abs/2302.05733.

Kassner, Nora and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818. https://doi.org/10.18653/v1/2020.acl-main.698

Kew, Tannon, Florian Schottmann, and Rico Sennrich. 2023. Turning English-centric LLMs into polyglots: How much multilinguality is needed? *ArXiv preprint*, abs/2312.12683.

Khashabi, Daniel, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. 2022. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643. https://doi.org/10.18653/v1/2022.naacl-main.266

Khashabi, Daniel, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907. https://doi.org/10.18653/v1/2020.findings-emnlp.171

Kim, Joohyun and Raymond Mooney. 2012. Unsupervised PCFG induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 433–444.

Kim, Seungone, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The CoT collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 12685–12708. https://doi.org/10.18653/v1/2023.emnlp-main.782

Kitaev, Nikita and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686. https://doi.org/10.18653/v1/P18-1249

Köksal, Abdullatif, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. LongForm: Optimizing instruction tuning for long text generation with corpus extraction. *ArXiv preprint*, abs/2304.08460.

Köpf, Andreas, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul E. S., Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. OpenAssistant conversations—democratizing large language model alignment. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, pages 47669–47681.

Krishnamurthy, Jayant and Thomas Kollar. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 1:193–206. https://doi.org/10.1162/tacl_a_00220

Kuhlmann, Gregory, Peter Stone, Raymond Mooney, and Jude Shavlik. 2004. Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer. In *The AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*, pages 2468–2470.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. https://doi.org/10.1109/5.726791

Lester, Brian, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. https://doi.org/10.18653/v1/2021.emnlp-main.243

Li, Bangzheng, Wenpeng Yin, and Muhao Chen. 2022. Ultra-fine entity typing with indirect supervision from natural language inference. *Transactions of the Association for Computational Linguistics*, 10:607–622. https://doi.org/10.1162/tacl_a_00479

Li, Bo, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2023a. MIMIC-IT: Multi-modal in-context instruction tuning. *ArXiv preprint*, abs/2306.05425.

Li, Judith Yue, Aren Jansen, Qingqing Huang, Ravi Ganti, Joonseok Lee, and Dima Kuzmin. 2022a. MAQA: A multimodal QA benchmark for negation. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, pages 135–156.

Li, Toby Jia Jun, Tom Mitchell, and Brad Myers. 2020. Interactive task learning from GUI-grounded natural language instructions and demonstrations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 215–223. https://doi.org/10.18653/v1/2020.acl-demos.25

Li, Xiaonan and Xipeng Qiu. 2023. Finding supporting examples for in-context learning. *ArXiv preprint*, abs/2302.13539. https://doi.org/10.18653/v1/2023.findings-emnlp.411

Li, Xiang Lisa and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597. https://doi.org/10.18653/v1/2021.acl-long.353

Li, Yafu, Yongjing Yin, Jing Li, and Yue Zhang. 2022b. Prompt-driven neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2579–2590. https://doi.org/10.18653/v1/2022.findings-acl.203

Li, Zekun, Baolin Peng, Pengcheng He, and Xifeng Yan. 2023b. Do you really follow me? Adversarial instructions for evaluating the robustness of large language models. *ArXiv preprint*, abs/2308.10819.

Lialin, Vladislav, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *ArXiv preprint*, abs/2303.15647.

Liang, Percy, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99. `https://doi.org/10.3115/1687878.1687893`

Lin, Chin Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Lin, Xi Victoria, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. RA-DIT: Retrieval-augmented dual instruction tuning. *ArXiv preprint*, abs/2310.01352.

Lin, Xi Victoria, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 9019–9052. `https://doi.org/10.18653/v1/2022.emnlp-main.616`

Liu, Haokun, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, pages 1950–1965.

Liu, Jiachang, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114. `https://doi.org/10.18653/v1/2022.deelio-1.10`

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35. `https://doi.org/10.1145/3560815`

Liu, Qian, Fan Zhou, Zhengbao Jiang, Longxu Dou, and Min Lin. 2023b. From zero to hero: Examining the power of symbolic tasks in instruction tuning. *ArXiv preprint*, abs/2304.07995.

Liu, Wei, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023c. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. *ArXiv preprint*, abs/2312.15685.

Liu, Xiao, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *ArXiv preprint*, abs/2103.10385.

Liu, Yilun, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, et al. 2023d. Automatic instruction optimization for open-source LLM instruction tuning. *ArXiv preprint*, abs/2311.13246.

Liu, Yixin, Alexander R. Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2023e. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. *ArXiv preprint*, abs/2311.09184.

Longpre, Shayne, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan Collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning, ICML 2023*, pages 22631–22648.

Lou, Renze and Wenpeng Yin. 2023. Forget demonstrations, focus on learning from textual instructions. *ArXiv preprint*, abs/2308.03795.

Lou, Renze, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu Su, and Wenpeng Yin. 2024. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations*.

Lu, Yao, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098. https://doi.org/10.18653/v1/2022.acl-long.556

Matuszek, Cynthia, Nicholas FitzGerald, Luke S. Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, pages 1435–1442.

Min, Sewon, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809. https://doi.org/10.18653/v1/2022.naacl-main.201

Min, Sewon, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064. https://doi.org/10.18653/v1/2022.emnlp-main.759

Mishra, Swaroop, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022a. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612. https://doi.org/10.18653/v1/2022.findings-acl.50

Mishra, Swaroop, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022b. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487. https://doi.org/10.18653/v1/2022.acl-long.244

Mishra, Swaroop and Elnaz Nouri. 2023. HELP ME THINK: A simple prompting strategy for non-experts to create customized content with models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11834–11890. https://doi.org/10.18653/v1/2023.findings-acl.751

Muennighoff, Niklas, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 15991–16111. https://doi.org/10.18653/v1/2023.acl-long.891

Murty, Shikhar, Pang Wei Koh, and Percy Liang. 2020. ExpBERT: Representation engineering with natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2106–2113. https://doi.org/10.18653/v1/2020.acl-main.190

Naik, Aakanksha, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353.

Nguyen, Tai and Eric Wong. 2023. In-context example selection with influences. *ArXiv preprint*, abs/2302.11042.

OpenAI. 2022. ChatGPT.

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

Ortiz, Jose Javier Gonzalez, John Guttag, and Adrian Dalca. 2023. Non-proportional parametrizations for stable hypernetwork learning. *ArXiv preprint*, abs/2304.07645.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, pages 27730–27744.

Pan, Alexander, Jun Shern Chan, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. 2023. Do the rewards justify the means? Measuring trade-offs between rewards and ethical behavior in the MACHIAVELLI benchmark. In *International Conference on Machine Learning, ICML 2023*, pages 26837–26867.

Parmar, Mihir, Swaroop Mishra, Mor Geva, and Chitta Baral. 2023. Don't blame the annotator: Bias already starts in the annotation instructions. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1779–1789. https://doi.org/10.18653/v1/2023.eacl-main.130

Peng, Baolin, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *ArXiv preprint*, abs/2304.03277.

Prasad, Archiki, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3845–3864. https://doi.org/10.18653/v1/2023.eacl-main.277

Pratap, Vineel, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. 2020. Massively multilingual ASR: 50 languages, 1 model, 1 billion parameters. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association*, pages 4751–4755. https://doi.org/10.21437/Interspeech.2020-2831

Press, Ofir, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5687–5711. https://doi.org/10.18653/v1/2023.findings-emnlp.378

Puig, Xavier, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. VirtualHome: Simulating household activities via programs. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 8494–8502. https://doi.org/10.1109/CVPR.2018.00886

Qiao, Shuofei, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 5368–5393. https://doi.org/10.18653/v1/2023.acl-long.294

Qin, Guanghui and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212. https://doi.org/10.18653/v1/2021.naacl-main.410

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Raheja, Vipul, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEDIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291. https://doi.org/10.18653/v1/2023.findings-emnlp.350

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. https://doi.org/10.18653/v1/D16-1264

Saha, Swarnadeep, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *ArXiv preprint*, abs/2310.15123.

Sainz, Oscar, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2439–2455. https://doi.org/10.18653/v1/2022.findings-naacl.187

Sainz, Oscar, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212. https://doi.org/10.18653/v1/2021.emnlp-main.92

Sanh, Victor, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Schick, Timo and Hinrich Schütze. 2021a. 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. https://doi.org/10.18653/v1/2021.eacl-main.20

Schick, Timo and Hinrich Schütze. 2021b. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402. https://doi.org/10.18653/v1/2021.emnlp-main.32

Schick, Timo and Hinrich Schütze. 2021c. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352. https://doi.org/10.18653/v1/2021.naacl-main.185

Song, Chiyu, Zhanchao Zhou, Jianhao Yan, Yuejiao Fei, Zhenzhong Lan, and Yue Zhang. 2023. Dynamics of instruction tuning: Each ability of large language models has its own growth pace. *ArXiv preprint*, abs/2310.19651.

Sorensen, Taylor, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 819–862. https://doi.org/10.18653/v1/2022.acl-long.60

Srivastava, Shashank, Igor Labutov, and Tom Mitchell. 2017. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1527–1536. https://doi.org/10.18653/v1/D17-1161

Srivastava, Shashank, Igor Labutov, and Tom Mitchell. 2018. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 306–316. https://doi.org/10.18653/v1/P18-1029

Stiennon, Nisan, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, pages 3008–3021.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650. https://doi.org/10.18653/v1/P19-1355

Su, Hongjin, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121. https://doi.org/10.18653/v1/2023.findings-acl.71

Sun, Lin, Kai Zhang, Qingyuan Li, and Renze Lou. 2024. UMIE: Unified multimodal information extraction with instruction tuning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014*, pages 19062–19070. https://doi.org/10.1609/aaai.v38i17.29873

Sun, Xiaobing and Wei Lu. 2022. Implicit n-grams induced by recurrence. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, pages 1624–1639. https://doi.org/10.18653/v1/2022.naacl-main.117

Suzgun, Mirac, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051. https://doi.org/10.18653/v1/2023s.findings-acl.824

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Tay, Yi, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations, ICLR 2023*.

Tellex, Stefanie, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, 32(4):64–76. https://doi.org/10.1609/aimag.v32i4.2384

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

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008.

Vogel, Adam and Daniel Jurafsky. 2010. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 806–814.

Wan, Alexander, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 35413–35425.

Wang, Liwen, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022a. InstructionNER: A Multi-Task Instruction-Based Generative Framework for Few-Shot NER. *ArXiv preprint*, abs/2203.03903.

Wang, Pei and Ben Goertzel. 2007. Introduction: Aspects of artificial general intelligence. In *Proceedings of the 2007 Conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006*, pages 1–16.

Wang, Xinyi, Wanrong Zhu, and William Yang Wang. 2023. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *ArXiv preprint*, abs/2301.11916.

Wang, Xuezhi, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023*.

Wang, Yizhong, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023b. How far can camels go? Exploring the state of instruction tuning on open resources. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, pages 74764–74786.

Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-Instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 13484–13508. https://doi.org/10.18653/v1/2023.acl-long.754

Wang, Yizhong, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Benchmarking generalization via in-context instructions on 1,600+ language tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Wang, Ziqi, Yujia Qin, Wenxuan Zhou, Jun Yan, Qinyuan Ye, Leonardo Neves, Zhiyuan Liu, and Xiang Ren. 2020. Learning from explanations with neural execution tree. In *8th International Conference on Learning Representations, ICLR 2020*.

Webson, Albert and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344. https://doi.org/10.18653/v1/2022.naacl-main.167

Wei, Jason, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, pages 24824–24837.

Wei, Jerry W., Le Hou, Andrew K. Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc V. Le. 2023. Symbol tuning improves in-context learning in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 968–979. https://doi.org/10.18653/v1/2023.emnlp-main.61

Weller, Orion, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375. https://doi.org/10.18653/v1/2020.emnlp-main.105

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Wu, Hui and Xiaodong Shi. 2022. Adversarial soft prompt tuning for cross-domain sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2438–2447. https://doi.org/10.18653/v1/2022.acl-long.174

Wu, Minghao, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2024. LaMini-LM: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers*, pages 944–964.

Wu, Zeqiu, Xiang Ren, Frank F. Xu, Ji Li, and Jiawei Han. 2018. Indirect supervision for relation extraction using question-answer pairs. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018*, pages 646–654. https://doi.org/10.1145/3159652.3159709

Wu, Zhiyong, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 1423–1436. https://doi.org/10.18653/v1/2023.acl-long.79

Xia, Congying, Wenpeng Yin, Yihao Feng, and Philip Yu. 2021. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1360. https://doi.org/10.18653/v1/2021.naacl-main.106

Xie, Jian, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024a. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Xie, Jian, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024b. TravelPlanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*.

Xu, Canwen, Daya Guo, Nan Duan, and Julian J. McAuley. 2023a. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 6268–6278. https://doi.org/10.18653/v1/2023.emnlp-main.385

Xu, Can, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023b. WizardLM: Empowering large language models to follow complex instructions. *ArXiv preprint*, abs/2304.12244.

Xu, Canwen, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. 2023c. Small models are valuable plug-ins for large language models. *ArXiv preprint*, abs/2305.08848.

Xu, Haike, Zongyu Lin, Jing Zhou, Yanan Zheng, and Zhilin Yang. 2023d. A universal discriminator for zero-shot generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 10559–10575. https://doi.org/10.18653/v1/2023.acl-long.589

Xu, Hanwei, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. 2022. ZeroPrompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4235–4252. https://doi.org/10.18653/v1/2022.findings-emnlp.312

Xu, Hanzi, Slobodan Vucetic, and Wenpeng Yin. 2022. OpenStance: Real-world zero-shot stance detection. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 314–324. https://doi.org/10.18653/v1/2022.conll-1.21

Xu, Wenda, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023e. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994. https://doi.org/10.18653/v1/2023.emnlp-main.365

Xu, Zhiyang, Ying Shen, and Lifu Huang. 2023. MultiInstruct: Improving multi-modal zero-shot learning via instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 11445–11465. https://doi.org/10.18653/v1/2023.acl-long.641

Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, pages 11809–11822.

Ye, Qinyuan, Xiao Huang, Elizabeth Boschee, and Xiang Ren. 2020. Teaching machine comprehension with compositional explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1599–1615. https://doi.org/10.18653/v1/2020.findings-emnlp.145

Ye, Qinyuan, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189. https://doi.org/10.18653/v1/2021.emnlp-main.572

Ye, Qinyuan and Xiang Ren. 2021. Learning to generate task-specific adapters from task description. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 646–653. https://doi.org/10.18653/v1/2021.acl-short.82

Ye, Seonghyeon, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2023. In-context instruction learning. *ArXiv preprint*, abs/2302.14691.

Ye, Seonghyeon, Joel Jang, Doyoung Kim, Yongrae Jo, and Minjoon Seo. 2022a. Retrieval of soft prompt enhances zero-shot task generalization. *ArXiv preprint*, abs/2210.03029.

Ye, Seonghyeon, Doyoung Kim, Joel Jang, Joongbo Shin, and Minjoon Seo. 2022b. Guess the instruction! Making language models stronger zero-shot learners. *ArXiv preprint*, abs/2210.02969.

Yin, Da, Xiao Liu, Fan Yin, Ming Zhong, Hritik Bansal, Jiawei Han, and Kai-Wei Chang. 2023a. Dynosaur: A dynamic growth paradigm for instruction-tuning data curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 4031–4047. `https://doi.org/10.18653/v1/2023.emnlp-main.245`

Yin, Wenpeng, Muhao Chen, Ben Zhou, Qiang Ning, Kai-Wei Chang, and Dan Roth. 2023b. Indirectly supervised natural language processing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 32–40. `https://doi.org/10.18653/v1/2023.acl-tutorials.5`

Yin, Wenpeng, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923. `https://doi.org/10.18653/v1/D19-1404`

Yin, Wenpeng, Jia Li, and Caiming Xiong. 2022. ConTinTin: Continual learning from task instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3062–3072. `https://doi.org/10.18653/v1/2022.acl-long.218`

Yu, Fei, Hongbo Zhang, and Benyou Wang. 2023. Nature language reasoning, a survey. *ArXiv preprint*, abs/2303.14725.

Yu, Zhaojian, Xin Zhang, Ning Shang, Yangyu Huang, Can Xu, Yishujie Zhao, Wenxiang Hu, and Qiufeng Yin. 2023. WaveCoder: Widespread and versatile enhanced instruction tuning with refined data generation. *ArXiv preprint*, abs/2312.14187.

Zeng, Aohan, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. GLM-130B: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations*.

Zhang, Hongming, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550. `https://doi.org/10.18653/v1/2020.emnlp-main.119`

Zhang, Kai, Bernal Jimenez Gutierrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812. `https://doi.org/10.18653/v1/2023.findings-acl.50`

Zhang, Shengyu, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *ArXiv preprint*, abs/2308.10792.

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022a. OPT: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

Zhang, Yi, Sujay Kumar Jauhar, Julia Kiseleva, Ryen White, and Dan Roth. 2021. Learning to decompose and organize complex tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2726–2735. `https://doi.org/10.18653/v1/2021.naacl-main.217`

Zhang, Yiming, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148. `https://doi.org/10.18653/v1/2022.emnlp-main.622`

Zhang, Zhuosheng, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Zhao, Zihao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, pages 12697–12706.

Zhong, Ruiqi, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878. `https://doi.org/10.18653/v1/2021.findings-emnlp.244`

Zhou, Chunting, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, pages 55006–55021.

Zhou, Denny, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Zhou, Jeffrey, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *ArXiv preprint*, abs/2311.07911.