Is Prompt All You Need? No.
A Comprehensive and Broader View of Instruction Learning

Renze Lou[†], **Kai Zhang**[◇] and **Wenpeng Yin**[†]
[†]The Pennsylvania State University; [◇]The Ohio State University
{renze.lou, wenpeng}@psu.edu; zhang.13253@osu.edu

Abstract

Task semantics can be expressed by a set of input-to-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 from task instructions. Despite its impressive progress, there are some common issues that the community struggles with. This survey paper tries to summarize the current research on instruction learning, particularly, by answering the following questions: (i) what is task instruction, and what instruction types exist? (ii) how to model instructions? (iii) what factors influence and explain the instructions' performance? (iv) what challenges remain in instruction learning? To our knowledge, this is the first comprehensive survey about textual instructions.

1 Introduction

One goal of AI is to build a system that can universally understand and solve new tasks. Labeled examples, as the mainstream task representation, are unlikely to be available in large numbers or even do not exist. Then, is there any other task representation that can contribute to task comprehension? Task instructions provide another dimension of supervision for expressing the task semantics.

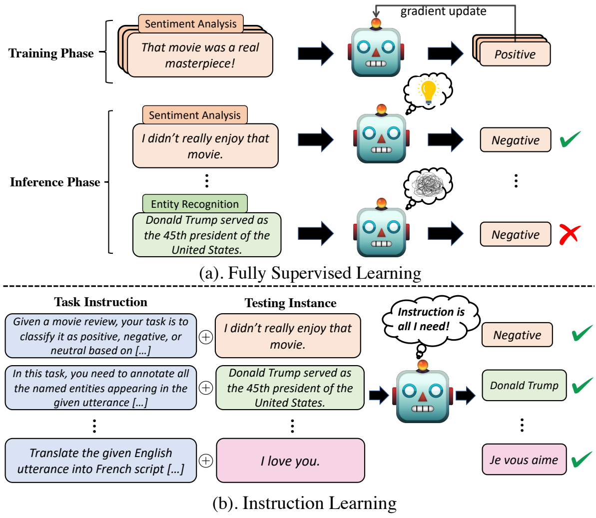

Figure 1: Two machine learning paradigms: (a) traditional fully supervised learning uses extensive labeled examples to represent the task semantics. It is expensive, and the resulting system hardly generalizes to new tasks; (b) instruction learning utilizes task instruction to guide the system quickly adapts to various new tasks.

Instructions often contain more abstract and comprehensive knowledge of the target task than individual labeled examples. As shown in Figure 1, with the availability of task instructions, systems can be quickly built to handle new tasks, especially when task-specific annotations are scarce. Instruction Learning is inspired by the typical human learning for new tasks, e.g., a little kid can well solve a new mathematical task by learning from its instruction and a few examples (Fennema et al., 1996; Carpenter et al., 1996). This new learning paradigm has recently attracted the main attention of the machine learning and NLP communities (Radford et al., 2019; Efrat and Levy, 2020; Brown et al., 2020, *inter alia*).

When talking about task instructions, most of us will first connect this concept with prompts–using a brief template to reformat new input into a language modeling problem so as to prime a PLM for a response (Liu et al., 2023). Despite prompts'

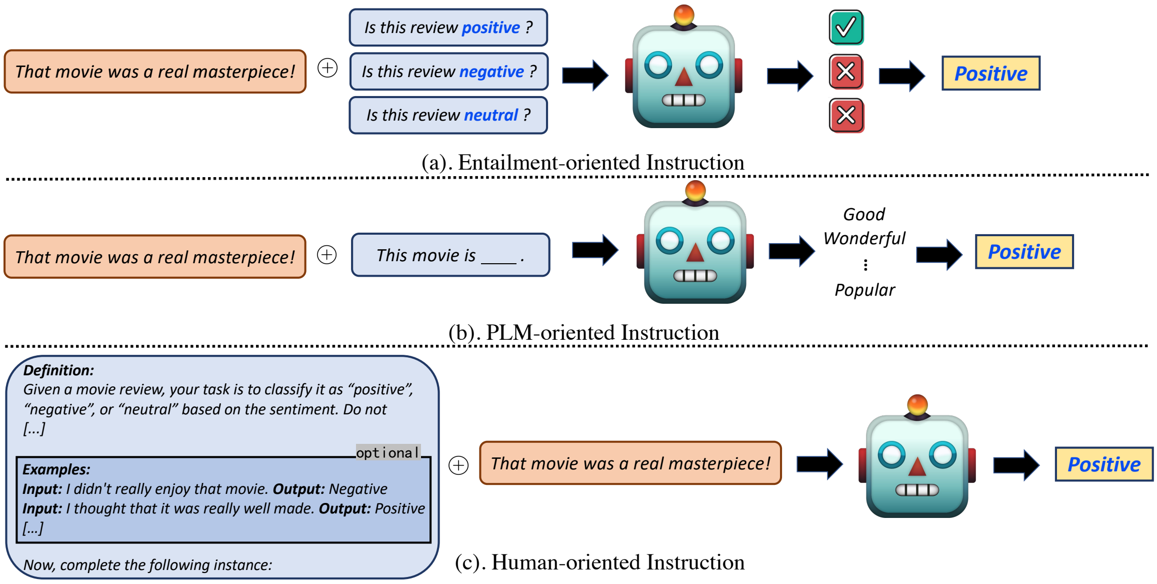

Figure 2: An illustration of three distinct categories of textual instructions. (a) **Entailment-oriented**: regarding the original input as a premise and converting each predefined label into a hypothesis (i.e., instruction). (b) **PLM-oriented**: using a template to construct the original task input into a cloze question. (c) **Human-oriented**: utilizing sufficient task information as instruction, such as definitions and optional few-shot demonstrations, etc.

prevalence in text classification, machine translation, etc., we claim that prompts are merely a special case of instructions. This paper takes a comprehensive and broader view of instruction-driven NLP research. Particularly, we try to answer the following questions:

- What is task instruction, and what instruction types exist? (§ 3)
- Given a task instruction, how to encode it to help the target task? (§ 4)
- What factors (e.g., model size, task numbers) impact the instruction-driven systems' performance, and how to design better instructions? (§ 5)
- What applications can instruction learning contribute? (§ 6)
- What challenges exist in instruction learning and what are future directions? (§ 7)

To our knowledge, this is the first paper that surveys textual instruction learning. In contrast to some existing surveys that focused on a specific in-context instruction, such as prompts (Liu et al., 2023), input-by-output demonstrations (Dong et al., 2023), or reasoning (Huang and Chang, 2022; Qiao et al., 2022), we provide a broader perspective to connect distinct researches in this area in an organized way. We hope this paper can present a better story of instruction learning and attract more peers to work on this challenging AI problem. We also release the corresponding reading list of this survey. [1]

2 Preliminary

For task instruction learning, we aim to drive the systems to reach the output given the input by following the instructions. Thus, a dataset 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 (e.g., 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.
- **Template** ($\hat{\mathrm{T}}$): a textual template that tries to express the task meaning on its own or acts as a bridge between X and Y.[2] $\hat{\mathrm{T}}$ may not be an instruction yet.

In § 3, we will elaborate that a task instruction I is actually a combination of $\hat{\mathrm{T}}$ with X or Y, or the $\hat{\mathrm{T}}$ on its own in some cases.

[1] https://github.com/RenzeLou/awesome-instruction-learning

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

task	TE premise (i.e., input text)	TE 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.

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

3 What is Task Instruction?

Various types of textual instructions have been used in previous zero- and few-shot NLP tasks, such as prompts (Hendrycks et al., 2021; Srivastava et al., 2022; Bach et al., 2022, *inter alia*), Amazon Mechanical Turk instructions (Mishra et al., 2022b; Wang et al., 2022c; Yin et al., 2022, *inter alia*), instructions augmented with demonstrations[3] (Khashabi et al., 2020; Ye et al., 2021; Min et al., 2022b, *inter alia*), and Chain-of-Thought explanations (Wei et al., 2022b; Lampinen et al., 2022; Li et al., 2022c, *inter alia*), etc. Different instructions are initially designed for distinct objectives (e.g., Mturk instructions are originally created for human annotators to understand, and prompts are to steer PLMs). In this section, as illustrated in Figure 2, we first summarize these instructions into three categories that perform different combinations of $\hat{\mathrm{T}}$, X, and Y (ENTAILMENT-ORIENTED, PLM-ORIENTED, and HUMAN-ORIENTED), then compare them and provide the formal definition of instructions.

3.1 I=$\hat{\mathrm{T}}$+Y: Entailment-oriented Instruction

One 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 focuses on encoding the input semantics while losing the label semantics. To let systems recognize new labels without relying on massive labeled examples, Yin et al. (2019) proposed to build 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 Table 1, this approach builds instructions (I) combining a template ($\hat{\mathrm{T}}$) with a label Y to explain each target label (Y). Since this paradigm naturally satisfies the format of textual entailment (TE, where the task inputs and the instructions can be treated as premises and hypotheses, respectively), these kinds of instructions are termed "Entailment-oriented Instructions".

The advantages of entailment-oriented instruction learning are four-fold: (i) it keeps the label semantics so that input encoding and output encoding both get equal attention in modeling the input-output relations; (ii) it results in a unified reasoning process—textual entailment—to handle various NLP problems; (iii) it creates the opportunity of making use of indirect supervision from existing TE datasets so that a pretrained TE model is expected to work on those target tasks without task-specific fine-tuning; (iv) it extends the original close-set labels classification problem into a recognization problem of open-domain open-form labels with few or even zero label-specific examples. Therefore, it has been widely used in a variety of few/zero-shot classification tasks, such as classifying topics (Yin et al., 2019), sentiments (Zhong et al., 2021), stances (Xu et al., 2022b), entity types (Li et al., 2022a), and entity relations (Murty et al., 2020; Xia et al., 2021; Sainz et al., 2021, 2022).

3.2 I=$\hat{\mathrm{T}}$+X: PLM-oriented Instruction (e.g., prompts)

The prompt is a representative of the PLM-oriented instructions, which is usually a brief utterance prepended with the task input (*prefix prompt*), or a cloze-question template (*cloze*

[3]We consider the templates used in the in-context learning as the instructions rather than the few-shot demonstrations.

Task	Input X	Template $\hat{\text{T}}$ (cloze question)	Answer	Output Y
Sentiment Classification	I would like to buy it again.	[X] The product is __	Great Wonderful . . .	Positive
Entity Tagging	[X1]: Donald Trump served as the 45th president of the United States from 2017 to 2021. [X2]: Donald Trump	[X1] [X2] is a __entity?	Politician President . . .	People
Relation Tagging	[X1]: Donald Trump served as the 45th president of the United States from 2017 to 2021. [X2]: Donald Trump [X3]: United States	[X1] [X2] is the __of [X3]?	President Leader . . .	The_president_of
Textual Entailment	[X1]: Donald Trump served as the 45th president of the United States from 2017 to 2021. [X2]: Donald Trump is a citizen of United States.	[X2]? __, because [X1]	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 des États-Unis . . .

Table 2: PLM-oriented instruction utilizes a task-specific template to convert the origin input into a fill-in-blank question. In most classification tasks, the intermediate answer should be further mapped into the predefined label.

prompt) [4]. It is basically designed for querying the intermedia responses (that can be further converted into the final answers) from the pre-trained LMs (PLM). Since the prompted input conforms to the pre-training objectives of PLM (e.g., the cloze-style input satisfies the masked language modeling objective (Kenton and Toutanova, 2019)), it help get rid of the reliance on the traditional supervised fine-tuning and greatly alleviates the cost of human annotations. Consequentially, prompt learning achieved impressive results on a multitude of previous few/zero-shot NLP tasks, e.g., question answering (Radford et al., 2019; Lin et al., 2021), machine translation (Li et al., 2022d), sentiment analysis (Wu and Shi, 2022), textual entailment (Schick and Schütze, 2021a,b), and named entity recognition (Cui et al., 2021; Wang et al., 2022a).

Despite the excellent performance of prompt techniques, there are still two obvious issues with PLM-oriented instructions in real-world applications: (i). ***Not User-Friendly***. As the prompt is crafted for service PLM, it is encouraged to design prompts in a "model's language" (e.g., model-preferred incoherent words or internal embedding). However, this PLM-oriented instruction is hard to understand and often violates human intuitions. (Gao et al., 2021; Li and Liang, 2021; Qin and Eisner, 2021; Khashabi et al., 2022). Meanwhile, the performance of prompts highly depends on the laborious prompt engineering (Bach et al., 2022), while most end-users are not PLM experts and usually lack sufficient knowledge to tune an effective prompt. (ii). ***Applications Constraints***. The prompt is usually short and simplistic, whereas many tasks can not 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., 2022).

[4]Please refer to Section 2.2.1 of Liu et al. (2023) for a detailed definition of prompt.

3.3 I=$\hat{\text{T}}$+ optional $\{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^k$: Human-oriented Instruction

Human-oriented instructions basically mean the instructions used for crowd-sourcing works on the human-annotation platforms (e.g., Amazon MTurk instructions). Different from the PLM-oriented instructions, human-oriented instructions are usually some human-readable, descriptive, and paragraph-style task-specific textual information, consisting of `task title`, `category`, `definition`, and `things to avoid`, etc. Therefore, human-oriented instructions are more user-friendly and can be ideally applied to almost any complex NLP task.

Accordingly, human-oriented instructions have attracted much more attention in recent years (Yin et al., 2022; Hu et al., 2022; Gupta et al., 2022; Longpre et al., 2023, *inter alia*). For example, Efrat and Levy (2020) tried to test whether GPT-2 (Radford et al., 2019) can follow the MTurk instructions to annotate some popular NLP datasets. Their results showed that HuggingFace's off-the-shelf GPT-2 (Wolf et al., 2019) worked poorly on following these human-oriented instructions. While recent works found that multi-task instruction fine-tuned LMs could get more positive results. For instance, Mishra et al. (2022b) collected more than 60 NLP tasks with the corresponding MTurk instructions; Wang et al. (2022c) further extended this collection into a 1.6k cross-lingual

Task	Input X	Demonstrations $\{X_i, Y_i\}_{i=1}^{k}$	Template $\hat{T}$	Output Y
Named Entity Extraction	The last three days have been incredible. Eric Lauren sqiddly and diddly all happy.	[X1]: Let us wait for Tal August for a while. [Y1]: Tal August [X2]: I would like to make a face-to-face interaction with Sachin in August. [Y2]: Sachin . . .	Definition: In this task, you will be given sentences . . . to recognize the name of a person . . . Examples: Input: [X1]; Output: [Y1] Input: [X2]; Output: [Y2] Now complete the following example: Input: [X]; Output: __	Eric Lauren
Summarization	A plane carrying U.S. Rep. Gabrielle Giffords departed Houston . . .	[X1]: The goals just keep flowing for Lionel Messi . . . [Y1]: Lionel Messi scores twice as Barcelona beat Mallorca . . . [X2]: U.S. officials say they have specific reasons . . . [Y2]: Chairman of the House Intelligence Committee complains about	Definition: In this task, you are given news articles, and you need to generate a highlight . . . Examples: Input: [X1]; Output: [Y1] Input: [X2]; Output: [Y2] Now complete the following example: Input: [X]; Output: __	James McBride wins for The Good Lord Bird . . .
Question Generation	Piedmont, or mountain, glaciers are found in many parts of the world . . .	[X1]: There are a million times more viruses on the planet than stars in the universe . . . [Y1]: How many segments of virus DNA does the human genome contain? [X2]: When mice are kept at high population densities . . . [Y2]: When does the mice behavior change? . . .	Definition: Given a paragraph, your job is to generate a question that can be answered from the passage . . . Examples: Input: [X1]; Output: [Y1] Input: [X2]; Output: [Y2] Now complete the following example: Input: [X]; Output: __	What mountain ranges are present in South America?

Table 3: Examples coming from Wang et al. (2022c). The human-oriented instruction is similar to the PLM-oriented instruction, which also utilizes a template to convert origin input (in red) into a cloze question. However, the task template itself contains informative task semantics, i.e., the formal task definition. Meanwhile, few-shot alternative task demonstrations are also provided (in blue).

tasks scale. They all concluded that, after the large-scale instruction fine-tuning, the text-to-text PLM, like BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) can generalize to the challenging unseen tasks by benefiting from the MTurk instructions.

Common: make use of indirect supervision

Difference: instance-wise instruction vs. task-wise instruction

4 How to Model Instructions?

In this section, we summarize several most popular modeling strategies for instruction learning. Overall, we introduce four different modeling schemes: For the earlier machine learning-based systems, the (i). **Semantic Parser-based** strategy was the commonly chosen method for encoding instructions; As the neural networks and the pre-trained language models emerged, the (ii). **Prompting Template-based** and the (iii). **Prefix Instruction-based** instruction learning became two highly favored paradigms; Recently, (iv). **HyperNetwork-based** method also garnered greater interest.

4.1 Semantic Parser-based

At the early stage of machine learning, to help the systems understand natural language instructions, a great number of works employed semantic parsing to convert the instruction into the formal language (logical formula), which can be easier executed by the systems (e.g., "*You can move any top card to an empty free cell*" → "*card(x)* ∧ *freecell(y)* ∧ *empty(y)*") (Matuszek et al., 2012; Babeş-Vroman et al., 2012; Chen, 2012; Goldwasser and Roth, 2014, *inter alia*).

For example, Kuhlmann et al. (2004) first tried to utilize natural language instructions to guide the systems to play soccer games, where they trained an individual semantic parser in advance, and then mapped the textual instructions into formal languages that can be used to influence the policy learned by the reinforcement learner. Since constructing a fully-supervised semantic parser requires laborious human annotations, the follow-up works also used indirect or weak supervision coming from the grounded environments (e.g., knowledge base) to train the semantic parser (Eisenstein et al., 2009; Chen and Mooney, 2008; Kim and Mooney, 2012; Artzi and Zettlemoyer, 2013; Krishnamurthy and Kollar, 2013). Besides using the converted formal languages to guide the systems to complete specific tasks, some works also utilized the logical formulae of the instructions to perform data and feature augmentations (Srivastava et al., 2017; Hancock et al., 2018; Wang et al., 2020; Ye et al., 2020).

4.2 Prompting Template-based

As for the neural network-based systems, we can directly encode the natural language instructions into the model's embedding without the help of a semantic parser. One of the prominent modeling strategies is using the prompting template. The essence of the prompting template-based approach is to use a template to convert the task input into a prompted format (i.e., cloze question [5]). According to the terminologies in § 3, the final input of LMs can be described as $x^{'} = f(x)$ or $f(x, I)$, i.e., the task-agnostic template $f(\cdot)$ is required but the task-specific information I is optional. For example, $x =$ "*I love this movie.*" and $x^{'} =$ "*I love this movie. It was* __", where there is no any task-specific information provided.

The prompting template-based approach is particularly useful for modeling PLM-oriented and entailment-oriented instructions. Therefore, a lot of previous works employed this strategy to modeling prompts (Petroni et al., 2019; Jiang et al., 2020; Cui et al., 2021; Haviv et al., 2021; Schick and Schütze, 2021c, *inter alia*). Besides using the discrete prompting template, recent works also tried to tune the continuous templates and achieved incredible results (Li and Liang, 2021; Tsimpoukelli et al., 2021; Han et al., 2021, *inter alia*).

4.3 Prefix Instruction-based

Distinct from the prompting template-based approach, the prefix instruction-based method is mainly used for modeling human-oriented instructions, where sufficient task-specific information is provided (Mishra et al., 2022b; Wang et al., 2022c; Yin et al., 2022; Gu et al., 2022, *inter alia*). Formally, in this case, the final input of LMs can be written as $x^{'} = I \oplus x$ or $f(x, I)$, which means the I is required and has to be the prefix of the task input x, but the template language $f(\cdot)$ is optional. For example, Wang et al. (2022c) utilized the following template "*Definition:* $\cdots$ *Input:* $\cdots$ *Output:*" as the $f(\cdot)$, where all the task-specific information I is prepended with the task input x.

4.4 HyperNetwork-based

There are two obvious problems in using the prefix instruction-based modeling strategy. First, it concatenates the task-level instruction with every instance-level input, the repeating procedure significantly slowing down the processing/inference speed and the lengthy input also increasing the burden of computational cost (Liu et al., 2022). Second, it can potentially impact the optimization because the model can not explicitly distinguish the task input x from the prefix instructions I, thus the model can simply learn to complete the task and ignore the instructions (Webson and Pavlick, 2022; Deb et al., 2022).

To address the above issues, recent works began to employ the hypernetwork (Ha et al., 2016) to encode the task instructions (Jin et al., 2020; Deb et al., 2022; Ivison et al., 2022). The essences of using hypernetwork-based approach are (i). encoding the task instruction I and the task input x separately, and (ii). converting the instruction into task-specific model parameters. For example, Ye and Ren (2021) used the hypernetwork to convert the task instruction into several parameter-efficiency adaptors (Houlsby et al., 2019). Then they inserted these adaptors behind the multi-head attention layers of the underlying LMs to perform cross-task generalization.

5 Analyses

Instruction learning is proven to be effective in a lot of zero- and few-shot NLP tasks, but how to explain the impressive performance of instruction? And which aspects make a successful instruction learning procedure? To figure out the empirical rules in using instructions and better understand instruction learning, in this section, we summarize some insights from existing works for further research, i.e., some important factors that contribute to cross-task generalization.

5.1 Instruction-based Fine-tuning

We first emphasize the importance of instruction-based fine-tuning (a.k.a. instruction fine-tuning) [6].

[5] Unlike the definition in Liu et al. (2023), we regard any fill-in-blank question as the cloze, no matter whether the blank is in the middle or at the end of the question.

[6] We use the term "instruction fine-tuning" to distinguish it from "instruction tuning", e.g., tuning soft prompts.

Different from the traditional fine-tuning, which aims at training model to complete specific tasks (i.e., $x \rightarrow y$), instruction fine-tuning trains the LMs on various instruction datasets where each input is converted into an instruction style by using the prompting template $f(\cdot)$ (i.e., either prefix instruction or cloze instruction (Liu et al., 2023)), to drive the models to learn to follow the instruction (i.e., $f(x, I) \rightarrow y$). Existing works demonstrate that multi-task instruction-tuned LMs could better follow the instructions of the unseen tasks compared with no-tuned LMs (Wei et al., 2022a; Sanh et al., 2022; Yin et al., 2022; Chung et al., 2022; Prasad et al., 2022, *inter alia*).

However, since previous works demonstrated that a massive multi-task training procedure also benefits the downstream tasks learning of LMs (McCann et al., 2018; Aghajanyan et al., 2021; Aribandi et al., 2022), there is always a question that "*Whether instruction fine-tuning or multi-task learning plays a key role in cross-task generalization*". To answer this question, we first introduce the work of Weller et al. (2020), who solely tuned LMs with a multi-task learning paradigm and discovered that the LMs could find it hard to follow the instructions of the unseen tasks. Wei et al. (2022a); Sanh et al. (2022) further conducted in-depth comparison between multi-task instruction fine-tuning and multi-task learning on cross-task generalization. They found that instruction fine-tuning is the key for cross-task generalization rather than the multi-task learning itself.

Besides the performance gains on the unseen tasks, there are also many other benefits of instruction fine-tuning. For example, Wei et al. (2022a) showed that instruction fine-tuned LMs performed better on following the soft instructions. Meanwhile, Longpre et al. (2023) compared the convergences of Flan-T5 and T5 on single-task fine-tuning, and they found that instruction fine-tuned Flan-T5 learnt faster than T5 on the downstream single-task fine-tuning. What's more, some works also found that instruction fine-tuning makes LMs robust to some tiny perturbations in the instructions, such as the wordings (Sanh et al., 2022) and the paraphrasing (Gu et al., 2022). While off-the-shelf LMs are usually sensitive to the small instruction perturbations (Efrat and Levy, 2020; Weller et al., 2020), thus they require laborious prompt engineering (Bach et al., 2022). All in all, instruction fine-tuning tames the LMs to become much more user-friendly (Chung et al., 2022).

5.2 Instruction Paradigm

Keeping the instruction paradigm is also a crucial factor in instruction learning, which mainly means retaining the instruction conciseness of tuning and testing. Wei et al. (2022a) first investigated the performance impact of changing instruction paradigm. And they found that, the performances of the LMs tuned with short task names dropped when evaluating with more extended sentence-style instructions (i.e., task names $\rightarrow$ sentence-style instructions), compared with the results of keeping the instruction paradigm (i.e., task names $\rightarrow$ task names, or sentence-style instructions $\rightarrow$ sentence-style instructions). Meanwhile, Gu et al. (2022) also observed the performance dropping when changing paragraph-style instructions to sentence-style instructions in testing, further enhancing this conclusion.

Besides discrete instruction, the instruction paradigm is also critical in soft instruction learning. For example, Xu et al. (2022a) showed that the LMs fine-tuned with a continuous instruction also require a same-size prefix embedding when testing on unseen tasks, even if this embedding is randomly initialized. Interestingly, similar results were also found in the few-shot demonstrations (i.e., in-context learning). For instance, Min et al. (2022c) concluded that breaking the demonstration paradigm (i.e., removing the y) significantly harm the performance of MetaICL (Min et al., 2022b), which tuned with (x, y) pairs. Furthermore, Iyer et al. (2022) found that the number of demonstrations should also not be changed during evaluation (e.g., using 2 demonstrations in tuning and 3 in testing would result in the lower performance, compared with using 2 demonstrations in testing).

5.3 Model and Task Scale

A series of recent works demonstrated that the scale also matters in instruction learning, including both model parameters and tuning tasks [7] (Wei et al., 2022a; Sanh et al., 2022; Mishra et al., 2022b; Wang et al., 2022c; Xu et al., 2022a; Deb et al., 2022; Prasad et al., 2022; Chung et al., 2022; Iyer et al., 2022; Longpre et al., 2023, *inter alia*). A representative work among them is Chung et al. (2022), who conducted extensive experi-

[7] Task scale includes numbers and categories of tasks.

ments with 1,836 tuning tasks and 540B models. The results illustrated that the cross-task performance takes advantages from the both factors [8], suggesting the research community to continue scaling the instruction learning. However, the scale is usually unaffordable for most groups, and it also leads huge carbon emissions (Strubell et al., 2019; Schick and Schütze, 2021c), making it unrealistic in the real-world scenarios. Accordingly, some recent works began to investigate a more efficient way to address the scale problem, such as the parameter-efficient fine-tuning (Schick and Schütze, 2021a; Liu et al., 2022). For example, Jang et al. (2023) employed the instruction-level experts (adaptors) to fine-tuning LMs on a single task, which outperformed the multi-task fine-tuned LMs on the unseen tasks. It is also noteworthy that Longpre et al. (2023) adapted the idea of "noisy channel" (Min et al., 2022a) that extended the tuning task scale by simply inverting the input-output of instance and achieved a reasonable performance improvement.

5.4 Instruction Diversity

Instruction diversity (e.g., various creativity) at fine-tuning phase also affects the cross-task performance and robustness of LMs (Chung et al., 2022; Longpre et al., 2023). Notably, Sanh et al. (2022) fine-tuned T5 (Raffel et al., 2020) model on the multi-task datasets, where each dataset is associated with various instructions collected from "Public Pool of Prompts" (P3) (Bach et al., 2022). These instructions are written in different ways and perspectives (but in a similar conciseness) [9]. By varying the number of instructions per dataset used in fine-tuning, Sanh et al. (2022) found that the model fine-tuned with more diverse instructions achieved better and more robust performance on the unseen tasks. What's more, Sanh et al. (2022) also found that the instruction diversity could compensate the limited model scale, i.e., a relative small LMs (T0-3B) could still benefit from multi-task fine-tuning due to the mixture of diverse instructions [10].

Since manually crafting instructions with diversity is expensive and usually hard to achieve (Huynh et al., 2021; Parmar et al., 2022), recent works also resorted to model-generated instructions (Zhang et al., 2020, 2021; Honovich et al., 2022a,b; Ye et al., 2022b; Bai et al., 2022). For instance, Wang et al. (2022b) tried to drive GPT-3 (Brown et al., 2020) to generate quantitative instructions from scratch iteratively. Although the self-generated instructions contain more noise, owing to their diverse verb-noun structures (Kitaev and Klein, 2018) and lengths, they could still bring benefits to tuning the GPT-3 and show complementary effects with the human-written instructions. These results imply the profitability of instruction diversity, even at the expense of the correctness of instructions. We will further discuss it in § 7.2.

5.5 Instruction Taxonomies and Situations

As we have introduced in § 3, there are several kinds of textual instructions. Although they were all widely adapted by the previous works, different taxonomies show various-degree effects. For example, existing works found that adding positive few-shot demonstrations in the textual instructions could lead to a significant performance improvement on the unseen tasks (Mishra et al., 2022b; Wang et al., 2022c; Yin et al., 2022; Deb et al., 2022; Gu et al., 2022), especially for the tasks occupying complex output space (Wei et al., 2022a). Surprisingly, Gu et al. (2022) further found that combining incorrect instructions with correct demonstrations could outperform using correct instruction without demonstrations, indicating the key role of demonstrations in instruction learning. This prominence is perhaps because the LMs prefer to exploit the more superficial aspects of the demonstrations rather than the other complex contents (cf. Min et al., 2022c).

In addition, the effectiveness of different instruction taxonomies also highly depends on the target evaluation tasks. For example, the concise cloze-style instructions are useful on tasks that can be naturally expressed as instructions (e.g., QA), while it seems to be redundant when facing with language modeling tasks (cf. Wei et al., 2022a). What's more, CoT explanations seem to be necessary only for tasks that require multi-steps reasoning (cf. Kojima et al., 2022). To this end, a practical suggestion is to mix different instruction taxonomies when tuning the LMs, which has been

[8] Worth noting that the benefits of the model scale seem to outweigh the task scale. Please refer to the Fig. 4 of Chung et al. (2022) and the follow-up work of Longpre et al. (2023).

[9] See Appendix G of Sanh et al. (2022) for more details.

[10] While Wei et al. (2022a) only used a fixed number of instructions and found that instruction fine-tuning harmed the performance of smaller LMs (Flan-8B).

proved to be efficient in tackling the various target evaluation tasks (Chung et al., 2022; Iyer et al., 2022; Longpre et al., 2023).

5.6 Model Preference

Another factor that can enhance the cross-task performance is making instructions conform to the *preference* of LMs, that is, converting the instructions into model-oriented styles.

Since the current instruction learning paradigm mainly employs the PLM as the backbone of the system, one of the potential explanations for why PLM-oriented instruction (i.e., prompt) can work is that prompt recalls the pre-training objective and activates the task-specific knowledge of the PLM. Some of the existing works demonstrated the importance of conforming to the pre-training objective of PLM when doing instruction fine-tuning (Tay et al., 2022). For example, Schick and Schütze (2021a,c) proposed the idea of *pattern exploit training* (PET), which used a prompt to convert the original task inputs into cloze-style questions and then fine-tuned the PLM on instruction datasets with the masked language modelling objective. They found that, taking advantage of recalling the pre-training objective, relatively small LMs, such as ALBERT (Lan et al., 2019), can outperform GPT-3 on the SuperGLUE benchmark (Wang et al., 2019). Furthermore, Iyer et al. (2022) found that the PLM could perform better on the unseen tasks after mixing a small proportion of pretraining-style data in the instruction fine-tuning dataset. Sanh et al. (2022); Wei et al. (2022a) also found that the PLM was more likely to fail at the tasks whose objective is different from the language modelling but improved by adopting cloze-style instructions. All these results are in line with the empirical rules of prompt engineering (Liu et al., 2023), which highlights the importance of aligning the prompts with the PLM [11].

Besides the objective of instruction fine-tuning, the way of designing instructions is also found critical. To better cater to the model's preference, recent works began employing continuous embedding (i.e., soft instructions) instead of human-understandable discrete instructions. (Lester et al., 2021; Liu et al., 2021; Ye et al., 2022a, *inter alia*). Similar conclusions are also found in the human-oriented instructions, where the PLM constantly fails at following the human-oriented instructions but gains significant improvements after reframing the instructions to cater to the model's preference (Mishra et al., 2022a; Prasad et al., 2022; Gonen et al., 2022; Deng et al., 2022; Wang et al., 2022b). Despite the performance profits, it is still controversial whether it is worthwhile to convert the original human-oriented instructions into PLM-oriented style, because it impairs the interpretability of instructions and is highly contrary to human intuition (Khashabi et al., 2022; Webson and Pavlick, 2022; Prasad et al., 2022). We will further discuss it in § 7.2.

6 Applications

6.1 Human-Computer Interaction

Textual instructions can be naturally regarded as a human-computer interaction method. Numerous previous works employed 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 et al., 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 et al., 2011; Babeş-Vroman et al., 2012; Goldwasser and Roth, 2014), generating real-time sports broadcast (Chen and Mooney, 2008; Liang et al., 2009), controlling software (Branavan et al., 2010), and querying external databases (Clarke et al., 2010), etc. Meanwhile, instructions are also widely adapted to help communicate with the system in solving NLP tasks, e.g., following instructions to manipulate strings (Gaddy and Klein, 2019), classifying emails based on the given explanations (Srivastava et al., 2017, 2018), and text-to-code generation (Acquaviva et al., 2021).

Recently, a growing body of research tended to design the human-computer communication procedure in an **iterative** and **modular** manner. For

[11] Using prefix prompts for the auto-regressive LMs, while using cloze prompts for the masked LMs. Please refer to Liu et al. (2023) for more details.

example, Li et al. (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. Similarly, Dwivedi-Yu et al. (2022) proposed a benchmark to iteratively instruct the PLM to improve the text, where each iteration only used a small piece of instruction with a precise purpose (e.g., "*Simplify the text*" or "*Make the text neutral*"). Besides, Chakrabarty et al. (2022) constructed a collaborative poem-writing system, where the user could initially provide an ambiguous instruction (e.g., "*Write a poem about cake*") and then incrementally refine the instruction with more details (e.g., "*Contain the word – 'chocolate'* ") by observing the model's intermediate outputs. Meanwhile, Mishra and Nouri (2022) proposed a biography generation system[12] that progressively collected the necessary personal information from the users (by asking questions in a dialogue scene to guide the users) and generated a paragraph-style bio finally. As it is usually hard for non-expert users to write a sufficient instruction in one shot, while 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 a more user-oriented system. Due to its practical values, we emphasize the importance of this branch of work in this paper.

6.2 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 (e.g., the sentence "*a very fair price*" is sentiment positive because *"the word 'price' is directly preceded by 'fair' "*). Therefore, some existing works also employed the instruction as a distant supervision to perform data or feature augmentation (Srivastava et al., 2018; Hancock et al., 2018; Ye et al., 2020). For instance, Srivastava et al. (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. While Wang et al. (2020) utilized the label explanations to annotate the raw corpus automatically and trained the classifier on the resulting noisy data.

Besides the straightforward augmentation, Su et al. (2022) further used the task instruction to enrich the 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.

6.3 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., 2022c; Chung et al., 2022, *inter alia*), the instruction is likely to become a breakthrough in achieving this ultimate goal.

Notably, the recent two remarkable applications of instructions, namely InstructGPT (Ouyang et al., 2022) and ChatGPT [13], also indicated a big step towards building generalist language models. However, unlike the other works that mainly employ instruction learning, ChatGPT also adopts some other components, e.g., reinforcement learning with human feedback (RLHF) [14]. Although the answer to "which component contributes more to the dramatic results of ChatGPT" remains ambiguous and needs further investigation, we intro-

[12] Mishra and Nouri (2022) actually experimented with more than 60 text generation tasks.

[13] `https://chat.openai.com/`

[14] At the time of writing, there is no published paper about ChatGPT. Thus, our discussion is mainly based on the underlying techniques of InstructGPT because they share similar philosophies. See OpenAI's blog for more details: `https://openai.com/blog/chatgpt`

duce some recent works to highlight the critical role of instruction learning. For example, Chung et al. (2022) conducted extensive experiments to evaluate the human-preference alignments of PaLM (Chowdhery et al., 2022). They found that, even without any human feedback, the instruction fine-tuning significantly reduced the toxicity in the open-ended generations of PaLM, such as gender and occupation bias. In addition, some other works also solely employed creative instructions instead of human feedback and achieved notable cross-task results (Bai et al., 2022; Honovich et al., 2022a; Wang et al., 2022b).

Although ChatGPT still suffers from many unsatisfactory aspects and is far from the generalist language model (Qin et al., 2023; Guo et al., 2023; Koco'n et al., 2023; Wang et al., 2023), we hope the goal of AGI can continue to be promoted by adopting and evolving more powerful techniques, including instruction learning.

7 Challenges and Future Directions

7.1 Negated Instruction Learning

Negation is a common linguistic property and has been found to be crucial for various NLP tasks, e.g., textual entailment (Naik et al., 2018; Kassner and Schütze, 2020). Specific to instruction learning, negation denotes any *things-to-avoid* information of in-context instructions, such as negated task descriptions and negative demonstrations. Although human can benefit a lot from the negation (Dudschig and Kaup, 2018), existing works found LMs often fails at following the negated instructions (Mishra et al., 2022b; Li et al., 2022b; Jang et al., 2022). For example, Mishra et al. (2022a) conducted error analyses on GPT-3 and found GPT-3 constantly unable to understand the negated task constraints in the MTurk instructions. Wang et al. (2022c) further found that adding negative demonstrations and explanations to the instructions could even harm the cross-task generalization performance of PLM.

Since negation has increasingly become a challenge in instruction learning, we provide several hints to inspire future work. One potential solution to utilize the negated instruction is unlikelihood training (Hosseini et al., 2021; Ye et al., 2022b), which trains the LMs to minimize the ground truth probability when negated instructions are conditioned. In contrast, Yin et al. (2022) proposed to pre-train 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., 2022) and n-gram representations (Sun and Lu, 2022), also provided insights into tackling this problem.

7.2 Explainable Instruction Learning

As we have mentioned in § 5, in order to achieve a promising cross-task performance, one of the critical factors is to convert the human-oriented instructions into a much more PLM-oriented format, i.e., making the instructions conform to the model's preference. Numerous previous works have verified the effectiveness of catering to the model's preference in designing instructions, such as using the model's perplexity in choosing appropriate instructions (Gonen et al., 2022). Despite the performance gains of the PLM-oriented instruction selection, the resulting instructions consistently violate human intuitions, questioning the reliability of PLM-oriented instruction (Webson and Pavlick, 2022). For example, Prasad et al. (2022) tried to rephrase the human-oriented instructions by using performance rewards as the criterion. Surprisingly, the resulting instructions that yield better performance are constantly semantically incoherent, task-irrelevant, or even misleading instructions. Similar results are also found in Khashabi et al. (2022), which mapped the continuous instructions back into the discrete space and found those effective instructions are usually associated with semantic-irrelevant utterances. 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, such as setting an additional criterion during the instruction optimization.

7.3 Explicit Instruction Learning

As we have introduced in § 5, multi-task instruction fine-tuning is a fundamental factor in the current instruction learning paradigm. Obviously, there are two issues in such a learning paradigm. (i). It relies on training on the massive labeled examples to learn the instructions, which is still expensive and unrealistic for using large-scale LMs; (ii). Although the ultimate goal of instruction-based fine-tuning is learning to follow instructions by observing various training tasks, the current training objective is still the maximum likelihood of traditional generation tasks. This implicit instruction learning objective can lead to sub-optimal optimization (i.e., LMs can easily learn to complete specific training tasks).

To this end, one desired future direction is to evolve a new training objective that can help LMs explicitly learn from instructions, which can alleviate the reliance on large-scale training instances. Moreover, a more ambitious and challenging idea is to drive the system to follow instructions without additional training on any labeled example of any specific task, which is similar to the conventional semantic parser-based paradigm (§ 4).

7.4 Scalable Oversight: A New Evaluation Paradigm for Generalist AI Systems

The evaluation procedure of the current research paradigm basically follows two steps: First, driving the systems to complete specific tasks; Second, using some automatic metrics to evaluate the systems. While in the context of evaluating advanced instruction learning systems (i.e., generalist language models), this traditional paradigm suffers from two issues: (i). Limited by the automatic metrics, we will find it hard to measure the progress of the system if the system has already been more capable than non-expert humans on these tasks; (ii). We could have no idea whether the system is a qualified generalist "assistant" for non-expert humans.

Accordingly, recent works proposed the idea of *scalable oversight* (Cotra, 2021; Bowman et al., 2022), which denoted a new research paradigm for appraising the generalist language models, including the following steps: (i). **Task Choices**. Choosing the tasks where the LMs can outperform the non-experts but underperform the experts; (ii). **Non-experts Annotation**. Instead of driving the model to complete the tasks, ask the non-experts to annotate the challenging tasks by acquiring useful information from the untrustworthy LMs, i.e., the LMs need to follow some general instructions (w/o any domain-specific prompts) of non-experts to provide assistance; (iii). **Experts Evaluation**. At the end of the experiments, ask the experts to evaluate the annotation correctness of non-experts. In doing so, we can continue to promote the progress of generalist LMs by aligning the highly capable LMs with domain experts. Meanwhile, we simulate a real-world application scenery for most non-expert users, where the generalist LMs play the role of an assistant w/o any domain-specific knowledge aided. By adopting this paradigm, Bowman et al. (2022) found that the non-experts can outperform both LMs-alone or human-alone results by benefiting from the assistance of LMs.

Overall, the scalable oversight paradigm can help future research to test whether current LMs (e.g., ChatGPT) can effectively assist non-expert users in solving challenging tasks.

8 Conclusion

In this survey, we comprehensively summarize numerous existing pieces of literature about instruction learning and provide a systematic overview of this field, including different instruction taxonomies, modeling strategies, some critical aspects of using instructions in engineering, and several popular applications. We also emphasize some distinct challenges and the corresponding hints for future research. To our knowledge, this is the first extensive survey about instruction learning. In summary, we hope this survey can offer insights and inspiration for further in-depth research on instruction learning.

References

Samuel Acquaviva, Yewen Pu, Marta Kryven, Theodoros Sechopoulos, Catherine Wong, Gabrielle E Ecanow, Maxwell Nye, Michael Henry Tessler, and Joshua B Tenenbaum. 2021. Communicating natural programs to humans and machines. arXiv preprint arXiv:2106.07824.

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5799–5811.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q Tran, Dara Bahri, Jianmo Ni, et al. 2022. ExT5: Towards Extreme Multi-Task Scaling for Transfer Learning. In International Conference on Learning Representations.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. arXiv preprint arXiv:1907.05019.

Yoav Artzi 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.

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

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Févry, et al. 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.

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

Yonatan Bisk, 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.

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosuite, Amanda Askell, Andy Jones, Anna Chen, et al. 2022. Measuring progress on scalable oversight for large language models. arXiv preprint arXiv:2211.03540.

SRK Branavan, 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.

SRK Branavan, 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.

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

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. arXiv preprint arXiv:2212.03827.

Thomas P Carpenter, Elizabeth Fennema, and Megan L Franke. 1996. Cognitively guided instruction: A knowledge base for reform in primary mathematics instruction. The elementary school journal.

Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. arXiv preprint arXiv:2210.13669.

David Chen. 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.

David Chen and Raymond Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 25, pages 859–865.

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In Proceedings of the 25th international conference on Machine learning, pages 128–135.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction. In Proceedings of the ACM Web Conference 2022, pages 2778–2788.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling Language Modeling with Pathways. arXiv preprint arXiv:2204.02311.

Hyung Won Chung, 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 arXiv:2210.11416.

James Clarke, 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.

Ajeya Cotra. 2021. The case for aligning narrowly superhuman models. In AI Alignment Forum.

Leyang Cui, 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.

Budhaditya Deb, Guoqing Zheng, and Ahmed Hassan Awadallah. 2022. Boosting Natural Language Generation from Instructions with Meta-Learning. arXiv preprint arXiv:2210.11617.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RL-Prompt: Optimizing Discrete Text Prompts with Reinforcement Learning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3369–3391.

Qingxiu Dong, 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 arXiv:2301.00234.

Carolin Dudschig 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.

Jane Dwivedi-Yu, 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 arXiv:2209.13331.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? arXiv preprint arXiv:2010.11982.

Jacob Eisenstein, 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.

Elizabeth Fennema, 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.

David Gaddy 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.

Tianyu Gao, 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.

Ben Goertzel. 2014. Artificial General Intelligence: Concept, State of The Art, and Future Prospects. Journal of Artificial General Intelligence, 5(1):1.

Dan Goldwasser and Dan Roth. 2014. Learning from natural instructions. Machine learning, 94(2):205–232.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. arXiv preprint arXiv:2212.04037.

Jiasheng Gu, Hanzi Xu, Liangyu Nie, and Wenpeng Yin. 2022. Robustness of Learning from Task Instructions. arXiv preprint arXiv:2212.03813.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. arXiv preprint arXiv:2301.07597.

Prakhar Gupta, 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.

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. arXiv preprint arXiv:1609.09106.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. arXiv preprint arXiv:2105.11259.

Braden Hancock, 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.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. BERTese: Learning to speak to BERT. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 3618–3623.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In International Conference on Learning Representations.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022a. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor. arXiv preprint arXiv:2212.09689.

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2022b. Instruction Induction: From Few Examples to Natural Language Task Descriptions. arXiv preprint arXiv:2205.10782.

Arian Hosseini, 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.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In International Conference on Machine Learning, pages 2790–2799. PMLR.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A Smith, and Mari Ostendorf. 2022. In-context Learning for Few-shot Dialogue State Tracking. arXiv preprint arXiv:2203.08568.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards Reasoning in Large Language Models: A Survey. arXiv preprint arXiv:2212.10403.

Jessica Huynh, Jeffrey Bigham, and Maxine Eskenazi. 2021. A Survey of NLP-related Crowdsourcing Hits: What Works and What Does Not. arXiv preprint arXiv:2111.05241.

Hamish Ivison, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew Peters.

2022. HINT: Hypernetwork Instruction Tuning for Efficient Zero-Shot Generalisation. arXiv preprint arXiv:2212.10315.

Srinivasan Iyer, 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 arXiv:2212.12017.

Joel Jang, 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. arXiv preprint arXiv:2302.03202.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2022. Can large language models truly understand prompts? a case study with negated prompts. arXiv preprint arXiv:2209.12711.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. Transactions of the Association for Computational Linguistics, 8:423–438.

Tian Jin, 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.

Nora Kassner 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.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT, pages 4171–4186.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, et al. 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.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1896–1907.

Joohyun Kim 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.

Nikita Kitaev and Dan Klein. 2018. Constituency Parsing with a Self-Attentive Encoder. arXiv preprint arXiv:1805.01052.

Jan Koco'n, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydlo, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Koco'n, Bartlomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, P. Milkowski, Marcin Oleksy, Maciej Piasecki, Lukasz Radli'nski, Konrad Wojtasik, Stanislaw Wo'zniak, and Przemyslaw Kazienko. 2023. ChatGPT: Jack of All Trades, Master of None. arXiv preprint arXiv:2302.10724.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. arXiv preprint arXiv:2205.11916.

Jayant Krishnamurthy 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.

Gregory Kuhlmann, 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. San Jose, CA.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L McClelland, Jane X Wang, and Felix Hill. 2022. Can language models learn from explanations in context? arXiv preprint arXiv:2204.02329.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite Bert for Self-Supervised Learning of Language Representations. arXiv preprint arXiv:1909.11942.

Brian Lester, 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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880.

Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022a. Ultra-Fine Entity Typing with Indirect Supervision from Natural Language Inference. Transactions of the Association for Computational Linguistics, 10:607–622.

Judith Yue Li, Aren Jansen, Qingqing Huang, Ravi Ganti, Joonseok Lee, and Dima Kuzmin. 2022b. Maqa: A multimodal qa benchmark for negation. In NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022c. Explanations from large language models make small reasoners better. arXiv preprint arXiv:2210.06726.

Toby Jia-Jun Li, 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.

Xiang Lisa Li 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.

Yafu Li, Yongjing Yin, Jing Li, and Yue Zhang. 2022d. Prompt-Driven Neural Machine Translation. In Findings of the Association for Computational Linguistics: ACL 2022.

Percy Liang, Michael I 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.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. arXiv preprint arXiv:2112.10668.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. arXiv preprint arXiv:2205.05638.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Computing Surveys, 55(9):1–35.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. arXiv preprint arXiv:2103.10385.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. arXiv preprint arXiv:2301.13688.

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

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The Natural Language Decathlon: Multitask Learning as Question Answering. arXiv preprint arXiv:1806.08730.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy Channel Language Model Prompting for Few-shot Text Classification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5316–5330.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022b. 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.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022c. Rethinking the Role of Demonstrations: What Makes In-context Learning Work? arXiv preprint arXiv:2202.12837.

Swaroop Mishra, 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.

Swaroop Mishra, 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.

Swaroop Mishra and Elnaz Nouri. 2022. Help me think: A simple prompting strategy for non-experts to create customized content with models. arXiv preprint arXiv:2208.08232.

Shikhar Murty, Pang Wei Koh, and Percy Liang. 2020. Expbert: Representation engineering with natural language explanations. arXiv preprint arXiv:2005.01932.

Aakanksha Naik, 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.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.

Mihir Parmar, Swaroop Mishra, Mor Geva, and Chitta Baral. 2022. Don't blame the annotator: Bias already starts in the annotation instructions. arXiv preprint arXiv:2205.00415.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2022. GRIPS: Gradient-Free, Edit-Based Instruction Search for Prompting Large Language Models. arXiv preprint arXiv:2203.07281.

Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. 2020. Massively Multilingual ASR: 50 Languages, 1

Model, 1 Billion Parameters. arXiv preprint arXiv:2007.03001.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8494–8502.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Reasoning with Language Model Prompting: A Survey. arXiv preprint arXiv:2212.09597.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? arXiv preprint arXiv:2302.06476.

Guanghui Qin 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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. OpenAI blog.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the Limits of Transfer Learning with a Unified Text-To-Text Transformer. J. Mach. Learn. Res., 21(140):1–67.

Oscar Sainz, 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.

Oscar Sainz, 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.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In International Conference on Learning Representations.

Timo Schick 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.

Timo Schick 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.

Timo Schick 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.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models. arXiv preprint arXiv:2206.04615.

Shashank Srivastava, 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.

Shashank Srivastava, 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.

Emma Strubell, 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.

Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. 2022. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. arXiv preprint arXiv:2212.09741.

Xiaobing Sun 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.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying Language Learning Paradigms. arXiv preprint arXiv:2205.05131.

Stefanie Tellex, 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.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. Advances in Neural Information Processing Systems, 34:200–212.

Adam Vogel and Dan Jurafsky. 2010. Learning to follow navigational directions. In Proceedings of the 48th annual meeting of the association for computational linguistics, pages 806–814.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. Advances in neural information processing systems, 32.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. 2023. On the Robustness of ChatGPT: An Adversarial and Out-of-Distribution Perspective. arXiv preprint arXiv:2302.12095.

Liwen Wang, 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 arXiv:2203.03903.

Pei Wang 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.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-Instruct: Aligning Language Model with Self Generated Instructions. arXiv preprint arXiv:2212.10560.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022c. 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.

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

Albert Webson 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.

Jason Wei, Maarten Bosma, Vincent 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 International Conference on Learning Representations.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903.

Orion Weller, 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.

Thomas Wolf, 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 arXiv:1910.03771.

Hui Wu 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.

Congying Xia, Wenpeng Yin, Yihao Feng, and S Yu Philip. 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.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. 2022a. ZeroPrompt: Scaling Prompt-Based Pretraining to 1,000 Tasks Improves Zero-Shot Generalization. arXiv preprint arXiv:2201.06910.

Hanzi Xu, Slobodan Vucetic, and Wenpeng Yin. 2022b. OpenStance: Real-World Zero-Shot Stance Detection. In Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL), pages 314–324.

Qinyuan Ye, 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.

Qinyuan Ye, 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.

Qinyuan Ye 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.

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

Seonghyeon Ye, Doyoung Kim, Joel Jang, Joongbo Shin, and Minjoon Seo. 2022b. Guess the Instruction! Making Language Models Stronger Zero-Shot Learners. arXiv preprint arXiv:2210.02969.

Wenpeng Yin, 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.

Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. ConTinTin: Continual Learning from Task Instructions. In ACL, pages 3062–3072.

Hongming Zhang, 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.

Yi Zhang, 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.

Ruiqi Zhong, 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.