Aligning Large Language Models with Human: A Survey

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang
Lifeng Shang, Xin Jiang, Qun Liu
Huawei Noah's Ark Lab
{wangyufei44,zhongwanjun1,liliangyou,mifei2,zeng.xingshan,wenyong.huang}@huawei.com
{Shang.Lifeng,Jiang.Xin,qun.liu}@huawei.com

Abstract

Large Language Models (LLMs) trained on extensive textual corpora have emerged as leading solutions for a broad array of Natural Language Processing (NLP) tasks. Despite their notable performance, these models are prone to certain limitations such as misunderstanding human instructions, generating potentially biased content, or factually incorrect (hallucinated) information. Hence, aligning LLMs with human expectations has become an active area of interest within the research community. This survey presents a comprehensive overview of these alignment technologies, including the following aspects. (1) **Data collection**: the methods for effectively collecting high-quality instructions for LLM alignment, including the use of NLP benchmarks, human annotations, and leveraging strong LLMs. (2) **Training methodologies**: a detailed review of the prevailing training methods employed for LLM alignment. Our exploration encompasses Supervised Fine-tuning, both Online and Offline human preference training, along with parameter-efficient training mechanisms. (3) **Model Evaluation**: the methods for evaluating the effectiveness of these human-aligned LLMs, presenting a multifaceted approach towards their assessment. In conclusion, we collate and distill our findings, shedding light on several promising future research avenues in the field. This survey, therefore, serves as a valuable resource for anyone invested in understanding and advancing the alignment of LLMs to better suit human-oriented tasks and expectations. An associated GitHub link collecting the latest papers is available at https://github.com/GaryYufei/AlignLLMHumanSurvey.

1 Introduction

Foundational Large Language Models (LLMs) such as GPT-3 are pre-trained on a vast textual corpus with objectives to predict subsequent tokens. This process equips LLMs with world knowledge, facilitating the generation of coherent and fluent text in response to various inputs. Despite these strengths, foundational LLMs are not always adept at interpreting a wide range of instructions and can produce outputs that deviate from human expectations. Additionally, these models may produce biased content or invent (hallucinated) facts, which can limit their practical usefulness.

Therefore, recent NLP research efforts focus on empowering LLMs to understand instructions and to align with human expectations. Early methods for training LLMs to follow instructions primarily use task instruction sets, which are compiled by combining manually crafted task instruction templates with instances from standard NLP tasks. However, such approaches often fall short of capturing the intricacies of practical user instructions, as these instructions tend to originate from artificial NLP tasks designed to test specific aspects of machine capabilities. Real-world user instructions, on the other hand, are significantly more diverse and complex. As a result, OpenAI explored Supervised Fine-Tuning (SFT) of LLMs using instructions annotated by a diverse group of human users. Models developed through this process, such as InstructGPT (Ouyang et al., 2022) and ChatGPT [1], have demonstrated a marked improvement in understanding human instructions and solving complex tasks. To further enhance alignment, Ouyang et al. (2022) incorporate the Reinforcement Learning from Human Feedback (RLHF) approach, which involves learning from human preferences through a reward model trained with human-rated outputs.

There are challenges in alignment processes and the subsequent evaluation: (a) Collecting high-quality data for both SFT and RLHF stages can be costly and time-consuming. (b) The training strategies need to be optimized as SFT training is resource-consuming, and reinforcement learning in RLHF often lacks stability. (c) Evaluating LLMs comprehensively is challenging, as limited NLP

[1] https://chat.openai.com/

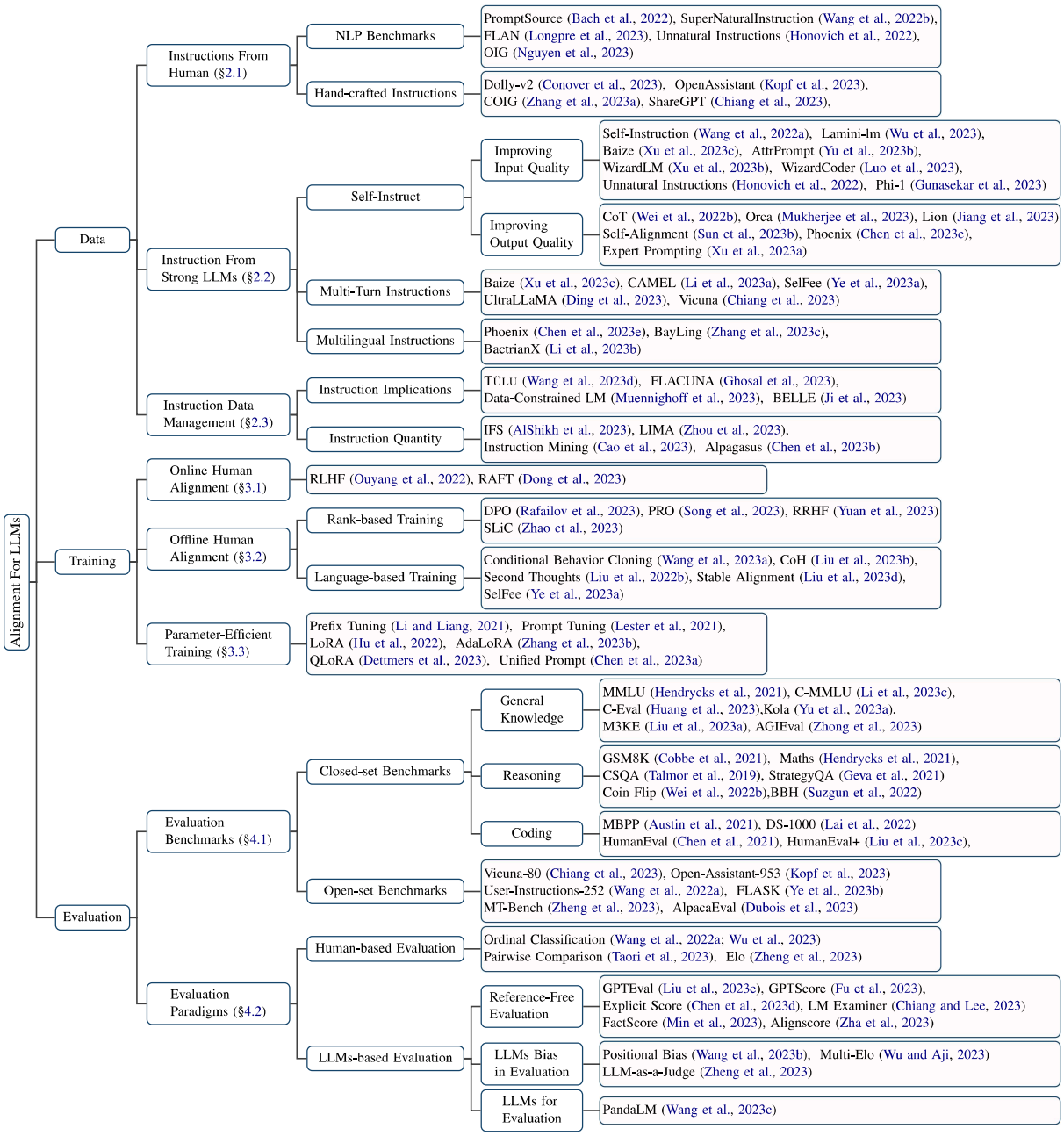

Figure 1: Taxonomy of research in aligning Large Language Models (LLMs) with human that consists of alignment data, training strategy, and evaluation methods.

benchmarks may not fully reveal the multifaceted capabilities of LLMs.

To address these limitations, extensive research efforts have been devoted. In Figure 1, we provide a summary of these multi-aspect approaches. For aspect (a), the focus is on effectively collecting large-scale, high-quality data for LLM alignment training. Researchers propose leveraging the power of existing NLP benchmarks, human annotators, and state-of-the-art LLMs (e.g., ChatGPT and GPT-4) to generate training instructions. To tackle aspect (b), solutions involve optimizing the training methods for better efficiency and stability in incorporating human preferences. Parameter-efficient training methods have been proposed to reduce computation burden and improve efficiency in LLM alignment. Additionally, some researchers consider human preference as ranking-based training signals or replace scalar rewards with language-based feedback to enhance training stability and performance. Regarding aspect (c), various human-centric LLM evaluation benchmarks and automatic evaluation protocols (e.g., LLMs for evaluation) have been proposed to obtain a comprehensive eval-

uation of aligned LLMs.

In this survey, we aim to provide a comprehensive overview of alignment technologies for large language models. In Section 2, we summarize various methods in effective high-quality data collection. Section 3 focuses on popular training methods to incorporate human preference data into LLMs. The evaluation benchmarks and automatic protocols for instruction-following LLMs are discussed in Section 4. By collating and distilling our findings, we shed light on several promising future research avenues in Section 5. Through this survey, we aim to provide an overview of the current state of LLM alignment, enabling researchers and practitioners to navigate the complexities of aligning LLMs with human values and expectations.

2 Alignment Data Collection

Aligning LLMs with human expectations necessitates the collection of high-quality training data that authentically reflects human needs and expectations. For the purposes of this survey, we conceptualize an instruction as $I_k = (x_k, y_k)$, where x_k denotes the instruction input and y_k denotes the corresponding response. This data can be derived from an array of sources, encompassing both human-generated instructions and those generated by strong LLMs. In this section, we summarize these methods of instruction generation and effective strategies for constructing a composite of diverse training instructions.

2.1 Instructions from Human

Human-provided instructions mainly originate from two main sources: pre-existing human-annotated NLP benchmarks and meticulously hand-crafted instructions.

2.1.1 NLP Benchmarks

An intuitive starting point for data collection involves adapting existing NLP benchmarks into natural language instructions. For instance, Figure 2 offers an example drawn from the Natural Language Inference task. Works such as PromptSource (Bach et al., 2022), FLAN (Wei et al., 2022a; Longpre et al., 2023), and SuperNaturalInstruction (Wang et al., 2022b; Mishra et al., 2022) are at the forefront of this approach. These benchmarks represent a substantial array of *diverse and heterogeneous* NLP tasks, such as dialogue, reasoning tasks and coding tasks, unified under the framework of language instructions. In each NLP

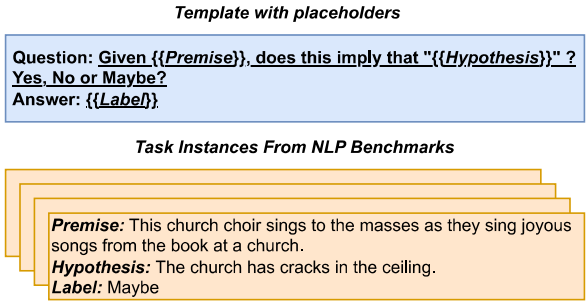

Figure 2: An Example of Instruction from a Natural Language Inference (NLI) benchmark.

benchmark, they engage annotators to craft several natural language templates that smoothly integrate all input data into a sequential text. The objective is to enhance LLMs' capability for multi-task learning across training tasks and foster generalization for unseen tasks. OIG (Nguyen et al., 2023) also combines instructions from FLAN-like NLP benchmarks with other types of open-ended instructions, such as how-to, maths and coding instructions. Concurrently, Honovich et al. (2022) put forth the concept of *Unnatural Instructions*, utilizing LLMs to generate new templates or instances bearing resemblance to the original instructions but with notable variances. Interestingly, the authors discovered that *text-davinci-002* outperforms GPT-3 in responding to these generated instructions, given that GPT-3 often devolved into repetitive or tangential outputs after providing the correct answer. This model of instruction creation is highly scalable and can yield millions of instructions effectively. Further, Wang et al. (2023d) demonstrated that FLAN-style instructions considerably enhanced the reasoning capabilities of aligned LLMs.

2.1.2 Hand-crafted Instructions

Constructing instructions from NLP benchmarks could be effective and painless. However, as many NLP datasets focus on a small and specific skill set, which means the resultant instructions are also relatively narrow in scope. Consequently, they may fall short in catering to the complex needs of real-world applications, such as engaging in dynamic human conversation.

To combat the above issues, it is possible to construct instructions via intentional manual annotations. How to effectively design a human-in-the-loop annotation framework becomes the key issue. The Databricks company collects a 15k crowd-sourcing instruction dataset *databricks-*

dolly-15k (Conover et al., 2023) from its employees. Those people are instructed to create prompt / response pairs in each of eight different instruction categories, including the seven outlined in Ouyang et al. (2022), as well as an open-ended free-form category. Importantly, they are *explicitly* instructed not to use external web information, as well as outputs from generative AI systems. Kopf et al. (2023) construct the *OpenAssistant* corpus with over 10,000 dialogues using more than 13,000 international annotators. The annotation process includes a) writing initial prompts for dialogue; b) replying as an assistant or user; c) ranking dialogue quality to explicitly provide human preferences. As a result, this corpus can be used for SFT and human preference alignment training for LLMs. Zhang et al. (2023a) construct high-quality Chinese instructions from existing English instruction datasets. They first translate the English instructions into Chinese, then verify whether these translations are usable. Finally, they hire annotators to correct and re-organize the instructions into the task description, input, output format in the selected corpus. ShareGPT [2], which is collected by Chiang et al. (2023), is an interesting exploration for crowd-sourcing human-written instructions. It is a website that encourages users to upload and share their interesting ChatGPT/GPT4 conversations. Such a mechanism can effectively collect a large number of diverse and human-written instructions that likely trigger high-quality ChatGPT/GPT4 responses. Popular online QA websites, such as Stack Overflow [3], Quora [4] and Zhihu [5], and large user-generated content databases, such as Wikipedia [6], are all reliable sources to provide high-quality human-written prompts for this purpose.Both Ding et al. (2023) and Xu et al. (2023c) propose to use these resources as the seed instructions to prompt GPT-3.5 to generate high-quality synthetic multi-turn dialogues.

2.2 Instructions From Strong LLMs

With the emergence of strong closed-source LLMs (e.g., ChatGPT/GPT4), it is also feasible to automate the collection process to obtain various types of synthetic instructions (e.g., single-turn, multi-turn, and multilingual instructions) by providing appropriate prompts to these LLMs. The main challenge is how to effectively prompt LLMs to generate diverse and high-quality instructions.

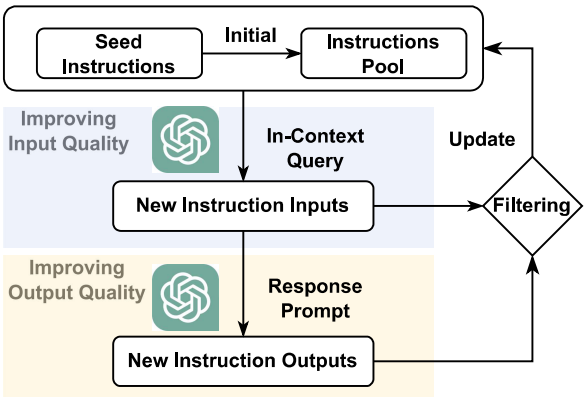

Figure 3: The overview of self-instruction. Starting from instructions in the pool, self-instruction leverages LLMs to produce new instructions via in-context learning. After filtering, LLMs are then prompted to respond to the remaining instructions. The full instructions are then added to the pool. Research efforts have been devoted to 1) Improving instruction input quality, and 2) Improving instruction output quality.

2.2.1 Self-Instruction

Self-Instruct (Wang et al., 2022a) were among the pioneers to automate the instruction collection process. It employed the in-context learning capability of ChatGPT to generate large-scale instructions from a pre-defined set of human-annotated instructions covering diverse topics and task types, as illustrated in Figure 3. The automatically generated instructions are followed by a quality control filtering process, and this iterative process continues until the desired data volume has been achieved. Interestingly, the researchers discovered that GPT-3 (Brown et al., 2020), fine-tuned with these instructions, performed better than models fine-tuned using instructions derived from NLP benchmarks SuperNI benchmark (Wang et al., 2022b) and *User-Oriented Instructions*, as discussed in Section 2.1). Several follow-up attempts, such as Aplaca (Taori et al., 2023) and its variants (Cui et al., 2023a) follow this *Self-Instruct* framework. More subsequent research efforts w.r.t. enhancing instruction diversity, quality, and complexity will be elaborated as follows.

Improving Input Quality One limitation of the synthetic instructions from strong LLMs often suffer from diversity issues. For example, Jentzsch and Kersting (2023) find that when prompting to

[2] https://sharegpt.com/
[3] https://stackoverflow.com/
[4] https://www.quora.com/
[5] https://www.zhihu.com/
[6] https://en.wikipedia.org/

generate jokes, ChatGPT only produces 25 unique joke patterns in thousands of samples. To improve the instruction input diversity, Wang et al. (2022a) propose different input and output generation strategies for different types of instructions. They first prompt ChatGPT to classify generated instruction into *classification tasks* or *non-classification tasks*. Then, they deploy output-first and input-first strategies for *classification tasks* and *non-classification tasks*, respectively. Others propose to add various external information into the input prompts to enhance diversity and factuality, including Wikipedia Category Keywords (Wu et al., 2023), user-generated questions on the Internet (e.g., Quora, StackOverflow) (Xu et al., 2023c; Anand et al., 2023) and instructions from the SuperNaturalInstruction benchmark (Honovich et al., 2022). Yu et al. (2023b) also shows that explicitly adding meta-information (e.g., length, topics, style) into the data generation prompts can effectively remove the bias in the generated synthetic data and improve the diversity of those synthetic data. Furthermore, Xu et al. (2023b) propose a novel *Evol-Instruct* framework to obtain complex and difficult instructions gradually. Instead of using existing instructions to prompt LLMs to produce new instructions via *in-context learning*, in *Evol-Instruct*, there are five different manually-designed prompts to explicitly instruct LLMs to rewrite the existing simple instructions into complex ones using in-depth methods (i.e., including more information on particular topics) or in-Breadth methods (i.e, improving topics/information coverage). The resulting WizardLM model is ranked top in the MT-Bench (Zheng et al., 2023) and AlpacaEval (Dubois et al., 2023). Luo et al. (2023) further expand this idea to produce complex code and programming instructions from the simple ones and propose the *WizardCoder* model, which outperforms several strong commercial LLMs, e.g., Anthropic's Claude and Google's Bard. Gunasekar et al. (2023) propose to generate textbook-like instructions prompted with sufficient background knowledge to promote reasoning and basic algorithmic skills of LLMs. They find that the resulting 1.3B LLMs *phi-1* successfully outperform various much larger LLMs, showing the importance of data quality.

Improving Output Quality Aside from the provision of high-quality instruction input, a critical requisite is to skillfully prompt LLMs to yield high-quality responses. The conventional method of enhancing response quality entails appending LLM prompts with additional conditions, encompassing the following facets.

(1) Reasoning-Provoking Conditions: Wei et al. (2022b) proposed the Chain-of-Thought (CoT) reasoning approach, which includes preconditions in the LLM prompts and generation the intermediate reasoning processes for complex problems, thereby assisting LLMs in problem-solving. Inspired by CoT, Mukherjee et al. (2023) developed the Orca model, which learns not only the superficial response text from LLMs, but also captures complex reasoning process signals. Specifically, they guided LLMs to respond to reasoning-intensive FLAN instructions with a series of predefined system prompts (e.g., "think step-by-step and justify your response"), spurring LLMs (e.g., GPT4) to disclose their reasoning process information. Thanks to these advancements, the Orca model significantly outperformed several powerful open-sourced LLMs.

(2) Hand-crafted Guiding Principles: Sun et al. (2023b) introduced *self-alignment* framework that incorporates 16 manually devised principle rules into input prompts, thereby steering LLMs towards generating useful, ethical, and reliable responses. To augment the impact of these rules, they employed the Chain-of-Thoughts (CoT) technology (Wei et al., 2022b), elucidating five examples to coach LLMs in discerning which rules to implement prior to generating actual response contents.

(3) Role-playing Conditions: Chen et al. (2023e) devised a method to generate a set of role profiles using a blend of ChatGPT and manual efforts. They created seed instructions for each role profile and applied *self-instruction* to the combination of role profiles and instructions to obtain nuanced responses from LLMs. Xu et al. (2023a) proposed a two-stage instruction response framework in which an expert profile is initially generated based on the instructions to be answered, followed by using both the expert profile and actual instructions to prompt LLMs for high-quality responses.

(4) Difficulty-monitoring Conditions: Jiang et al. (2023) proposed monitoring the quality of instruction response based on external LLM-based evaluations. They first fine-tune foundational LLMs with instruction data to obtain "student LLMs". Then, for each of training instruction, they gather responses from both teacher LLMs (e.g.,

ChatGPT) and student LLMs and prompted LLMs to conduct pairwise evaluation on the quality of both responses. Instructions are retained only when the student LLMs' response falls short of that from the teacher LLMs.

2.2.2 Multi-turn Instructions

In previous sections, we mainly focus on collecting synthetic single-turn instructions. However, LLMs well aligned with human should be capable to interact with users in a dialogue-based setting. To achieve this goal, some research efforts attempt to collect synthetic multi-turn instructions from strong LLMs. When aligning LLaMA with human, Vicuna (Chiang et al., 2023) leverage instructions from ShareGPT which is website hosting interesting human-LLMs joint conversations. However, ShareGPT requires large volumes of users to upload their conversations. Xu et al. (2023c) propose a novel Self-Chatting framework where questions from popular QA websites are used as the starting topics, then Chat-3.5 is prompted to chat with itself about this question in a four-turn dialogue. Li et al. (2023a) propose *CAMEL*, a "role-playing" framework where a human annotators first provide a topic, then LLMs are separately prompted to be "AI Users" and "AI Assistants" to discuss about this topic. Ji et al. (2023) take a step further and prompt LLMs to first determine the conversation topic and then ask LLMs to chat with themselves to produce dialogue corpus. Ye et al. (2023a) propose a novel revision-based multi-turn dialogue corpus. Specifically, after instructions and initial responses, they further prompt LLMs to generate feedback and the revised version of responses if necessary. They use this dataset to train the *SelFee* model and show that *SelFee* can effectively improve its own answers when prompted to do so without any external guidance. The UltraLLaMA model (Ding et al., 2023) leverages a wide range of real-world information, including (a) real-world knowledge from LLMs and Wikipedia; (b) various text creation tasks; (c) high-quality textual corpus, to produce initial questions and instructions that guide LLMs to generate diverse and high-quality multi-turn dialogues.

2.2.3 Multilingual Instructions

The above-generated instructions or dialogues are mostly based on English. To align LLMs with human who speak other languages, it is urgent and essential to expand the existing English resources into Multilingual ones. One straightforward idea is to translate instruction inputs and outputs into the target languages. Chen et al. (2023e) propose two translation strategies: *(a)* Post-answering which first translates the instruction inputs into the target language and then prompts strong LLMs to answer it. This could potentially preserve the specific culture patterns embedded in the target languages, but the output quality may be low as existing strong LLMs are often English-dominated; *(b)* Post-translating which first prompts strong LLMs to respond the instructions in English, then translate both inputs and outputs. This approach could obtain high-quality output text, but lost the specific culture information. Li et al. (2023b) follow the *Post-answering* strategy to construct instruction data for 52 popular languages using Google-Translate, then use these data to fine-tune LLaMA using the LoRA technology. An alternative solution is to mix several langauges in a multi-turn dialogue. BayLing (Zhang et al., 2023c) introduces a set of multi-turn *interactive translation* instructions to simultaneously improve multilingual and instruction-following ability for LLMs. Specifically, each multi-turn instruction is essentially a translation task where users first ask LLMs to translate a sentence to another language, then the users gradually add additional requirements (e.g., could you only use 10 words?). This process naturally connects different languages as well as human preferences with LLMs. We also summarize how to effectively adapt English-oriented LLMs to other languages in Appendix A.1.

2.3 Instruction Data Management

As discussed above, there are extensive approaches focusing on generating high-quality instructions from different sources. Naturally, it becomes critical to effectively manage all of these instruction data in the LLMs alignment.

Instruction Implications Several studies focus on the implications of instruction data. Ji et al. (2023) demonstrate that an increment in the total count of training instructions can be advantageous for standard NLP tasks (e.g., information extraction, classification, Closed QA, summarization). Yet, it bears negligible influence on complex reasoning tasks such as Math, Code, CoT, and Brainstorming. Intriguingly, Muennighoff et al. (2023) discover that adding approximately 50% of programming instructions not only leaves unaffected the general conversational performance but also

enhances the reasoning prowess of LLMs. In parallel, Ghosal et al. (2023) observe that integrating FLAN-style instructions with synthetic instructions from ChatGPT/GPT-4 effectively enhances LLMs' reasoning and problem-solving capacity.

Wang et al. (2023d) conduct a comprehensive analysis of the impacts of various instructions derived from different sources on factual knowledge, reasoning, coding, multilingual, and open-ended scenarios. They also reveal that instructions pertaining to CoT and Coding are vital for augmenting the reasoning capability of LLMs. Additionally, they ascertain that different instructions can affect different LLM capabilities. Therefore, a composite of all instruction types empowers the corresponding LLMs to reach their better overall performance, hinting at the need for more advanced instruction collection techniques and technologies.

Instruction Quantity Another critical question in instruction data management is the optimal quantity of instruction data required for effective LLM alignment. AlShikh et al. (2023) address this question by introducing a novel early-stopping criterion known as **IFS**. The premise of **IFS** rests on the observation that, given an input textual prefix, foundational LLMs typically predict ensuing tokens and generate "continuation-like" outputs, while fully instruction-tuned LLMs interpret the input prefix as questions, thereby generating "answer-like" outputs. **IFS** is quantified as the proportion of "answer-like" outputs within all its outputs given the instructions. The researchers train an external classifier to discriminate between "continuation-like" and "answer-like" outputs, concluding that LLaMA necessitates approximately 8K instructions to achieve a high IFS score. More instructions could potentially induce a semantic shift in the foundational LLMs. Zhou et al. (2023) similarly discern that merely 6K high-quality instructions suffice to align with human preferences. Motivated by these findings, researchers are investigating high-quality instruction selection. Cao et al. (2023) aim to identify predictive features of high-quality instructions. Initially, they extract representative features from the instruction dataset, then utilize these instructions to fine-tune LLMs. The feature importance is based on the model's performance. Their experiments demonstrate the better performance of LLMs trained on the resultant instructions. Differently, Chen et al. (2023b) propose using ChatGPT to directly assess the quality of instructions by assigning scores. They report that the LLM trained on the top 9K instructions notably outperforms those trained on the complete set of 52K Alpaca instructions.

3 Alignment Training

After collecting instructions from various sources, we then consider using these data to fine-tune existing foundational LLMs to align with human. The native solution is Supervised Fine-Tuning (SFT). Specifically, given instruction input x, SFT calculates the cross-entropy loss over the ground-truth response y as follows:

$$L_{ft} = -\sum_{t} \log P_{LLM}(y_{i',t}|x, y_{i',<t}) \quad (1)$$

Essentially, SFT helps LLMs to understand the semantic meaning of prompts and make meaningful responses. The main limitation of SFT is that it only teaches LLMs about the best responses and cannot provide fine-grained comparisons to sub-optimal ones. However, it is worth noting that SFT objective or SFT model parameters has also been integrated into many human preference training objective to regularize and stabilize the training process of LLMs. We summarize the research efforts built on top of SFT into: *Online human preference training*, *Offline human preference training* and *Parameter-effective fine-tuning solutions*.

3.1 Online Human Preference Training

Reinforcement learning from Human Feedback (RLHF) (Ouyang et al., 2022) is designed to learn the human preference signals from external reward models under the PPO framework. Specifically, RLHF consists of three main stages:

- **Step 1:** Collecting a high-quality instruction set and conducting SFT of pre-trained LLMs.
- **Step 2:** Collecting manually ranked comparison response pairs and train a reward model $\mathbb{R}$ to justify the quality of generated responses.
- **Step 3:** Optimizing the SFT model (policy) under the PPO reinforcement learning framework with reward calculated by $\mathbb{R}$.

In Step 3, to mitigate over-optimization issues, Ouyang et al. (2022) add a KL-divergence regularization between the current model weight and the SFT model weight obtained in Step 1. However, despite being effective in learning human preferences, PPO training is difficult in implementation

and stable training. Therefore, Dong et al. (2023) try to remove the PPO training in the above process and propose a novel Reward rAnked FineTuning (*RAFT*) method, which uses an existing reward model to select the best set of training samples based on the model outputs. Specifically, *RAFT* first samples a large batch of instructions, then uses the current LLMs to respond to these instructions. These data are then ranked by the reward model and only top $\frac{1}{k}$ instances are applied for SFT. *RAFT* can also be used in offline human preference learning where the global instruction set is continually updated with the top-ranked instructions in each batch. This contiguously updates the global instruction set to improve training data quality at each step.

3.2 Offline Human Preference Training

Although the above online algorithms have been shown effective in learning human preference, implementing these algorithms could be non-trivial because its training procedure requires interaction between policy, behavior policy, reward, and value model, which requires many hyper-parameters to be tuned to achieve better stability and performance. To avoid this issue, researchers also explore learning human preferences in an offline fashion.

3.2.1 Ranking-based Approach

As human preferences are often expressed as a ranking result over a set of responses, some research efforts directly incorporate the ranking information into the LLMs fine-tuning stage. Rafailov et al. (2023) propose *Direct Preference Optimization* (DPO), which implicitly optimizes the same objective as existing RLHF algorithms (i.e., reward function with a KL-divergence term) discussed above. Specifically, the DPO training objective can be written as:

$$\mathcal{L}_{\text{DPO}} = \log \sigma \left[\beta \log \left(\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{SFT}}(y_w \mid x)} \cdot \frac{\pi_{\text{SFT}}(y_l \mid x)}{\pi_\theta(y_l \mid x)} \right) \right] \tag{2}$$

where (x, y_w, y_l) is one instruction and two of the corresponding outputs with y_w ranked higher than y_l. Similarly, Song et al. (2023) propose *Preference Ranking Optimization* (PRO) method, an extended version of reward model training objective proposed in Ziegler et al. (2019), to further fine-tune LLMs to align with human preference. Given instruction x and a set of responses with human preference order $y^1 \succ y^2 \succ \cdots \succ y^n$, the objective can be defined as follows:

$$\mathcal{L}_{\text{PRO}} = -\sum_{k=1}^{n-1} \log \frac{\exp\left(\pi_\theta(y^k \mid x)\right)}{\sum_{i=k}^{n} \exp\left(\pi_\theta(y^i \mid x)\right)} \tag{3}$$

PRO also adds SFT training objective for the regularization purpose. Instead of adapting the reward training objective, Zhao et al. (2023) take the first step to calibrate the sequence likelihood using various ranking functions, including rank loss, margin loss, list rank loss (Liu et al., 2022c) and expected rank loss (Edunov et al., 2018). In addition, they also explore to use SFT training objective and KL-divergence as the regularization term. The experiment results on various text generation tasks show that the rank loss with the KL-divergence term performs the best. However, this paper only uses the BERTScore (Zhang* et al., 2020) between each candidate output and the ground-truth reference to simulate human preferences and they only conduct experiment on small pre-trained language models (i.e., no larger than 2B). Yuan et al. (2023) propose RRHF, which further optimizes LLaMA-7B to align with human preferences using a similar framework described above. RRHF is based on the list rank loss, but removes the margin terms based on the empirical results. In addition, different from Liu et al. (2022c), RRHF finds that the SFT training objective is more effective and efficient than KL-divergence in preventing LLMs from over-fitting. These results show that different ranking strategies should be adapted for LLMs with different size.

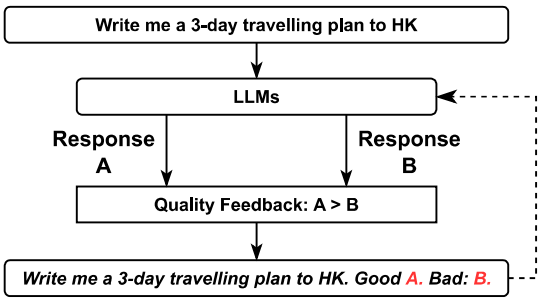

Figure 4: The overview of the Chain of Hindsigt (CoH) method. Responses with different quality are associated with different prefix. The CoH training loss is only applied on model output tokens (highlighted by red).

3.2.2 Language-based Approach

As reinforcement learning algorithms are hard to optimize and LLMs have strong text understanding ability, some works propose to directly use

natural language to inject human preference via SFT. Wang et al. (2023a) introduce the concept of "conditional behavior cloning" from offline reinforcement learning literature (Nguyen et al., 2022) to train LLMs to distinguish high-quality and low-quality instruction responses. Specifically, they design different language-based prefixes for different quality responses (e.g., high-quality response with "Assistant GPT4:" and low-quality response with "Assistant GPT3:"). This approach can effectively leverage both low- and high-quality training data to align LLMs with humans. Chain of Hindsight (CoH) (Liu et al., 2023b), on the other hand, directly incorporates human preference as a pair of parallel responses discriminated as low-quality or high-quality using natural language prefixes. As shown in Figure 4, after assigning human feedback to each model output, CoH concatenates the input instructions, LLMs outputs, and the corresponding human feedback together as the input to LLMs. Note that CoH only applies the fine-tuning loss to the actual model outputs, rather than the human feedback sequence and the instructions. During inference, CoH directly puts position feedback (e.g., good) after the input instructions to encourage the LLMs to produce high-quality outputs. It is worthnoting that, similar to Liu et al. (2022a); Ouyang et al. (2022), CoH also incorporates SFT objectives and random words masking to prevent LLMs from over-fitting.

Alternative approach is to explicitly incorporate revision-based instructions into LLMs training. Some preliminary studies have shown that many existing state-of-the-art LLMs have the capability to improve the quality of their responses when explicitly prompting them to do so (Chen et al., 2023c). Motivated by these findings, Liu et al. (2022b) recommend training LMs to produce edit operations between source (i.e., low-quality responses) and target (i.e., high-quality responses) sequences, which are subsequently integrated into a dynamic programming framework. Liu et al. (2023d) propose a novel type of instruction called *realignment*, designed to revise responses based on previously generated low-quality feedback and instructions. This compiled data is employed to instruct LLMs to self-correct when they generate bad responses. Similarly, Ye et al. (2023a) accumulate a multi-turn dialogue corpus utilizing this self-correction mechanism built with the ChatGPT models. Each dialogue starts with standard instructions, such as those from the Stanford Alpaca dataset. After ChatGPT has responded to the initial instructions, further revisions are explicitly requested until ChatGPT elects to terminate. They found that LLMs trained using these dialogues demonstrated an effective capacity to elevate the quality of their own responses.

3.3 Parameter-Effective Training

Directly fine-tuning all parameters in large language models (LLMs) would theoretically enable these models to adhere to provided instructions. However, this approach demands not only substantial computational resources, such as vast GPU memory but also extensive datasets for instruction training. In an effort to mitigate both computational and data requirements for constructing instruction-following LLMs, one potential route is the implementation of *Parameter-Effective Fine-tuning* strategies. Specifically, these methods froze the major part of LLM parameters and only train a limited set of additional parameters.

Supplementary Parameters Building upon this strategy, prefix tuning (Li and Liang, 2021) and prompt tuning (Lester et al., 2021) are inspired by the successful application of textual prompts in pre-trained language models (Brown et al., 2020). These methods either prepend trainable tokens to the input layer or each hidden layer, leaving the parameters of LLMs frozen during fine-tuning. Subsequently, He et al. (2022); Chen et al. (2023a) consolidated these strategies into unified frameworks, fostering more effective solutions for parameter-efficient fine-tuning.

Shadow Parameters While the above methodologies introduce supplementary parameters to LLMs, the following methods focus on training the weight representing model parameter variance without modifying the number of total model parameters during inference. For instance, Low-Rank Adaptation (LoRA) (Hu et al., 2022) suggests the addition of pairs of rank-decomposition trainable weight matrices (i.e., update matrices) to the existing weights, which are kept frozen. For example, given a neural layer $h = W_0 x$, LoRA modifies the forward pass as follows:

$$h = W_0 x + BAx \tag{4}$$

where $W_0 \in \mathbb{R}^{d\times k}$, $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, with the rank $r \ll \min(d, k)$. LoRA only updates the

parameters of A and B during training. Despite being effective, LoRA equally allocates parameter budgets over the whole LLMs, ignoring the varying importance of different weight parameters. Zhang et al. (2023b) propose AdaLoRA to combat this issue. Specifically, AdaLoRA first calculates the parameter importance using the training gradient and then determines the r values for different parameters matrix. Dettmers et al. (2023) propose QLoRA that further improves over LoRA by reducing memory usage, enabling a 65B LLM to be fine-tuned using a single 48G GPU. Specifically, QLoRA quantizes the transformer backbone model to 4-bit precision and uses paged optimizers to handle memory spikes.

Trade-offs For Parameter-efficient Training There are some successful applications of parameter-efficient training technologies, including the *Alpaca-LoRA* project [7], which is based on the Hugging Face's PEFT library (Mangrulkar et al., 2022) to train Alpaca using a single commercial GPU and Xu et al. (2023c), which apply LoRA to all linear layers in LLaMA to improve its adaption capabilities. However, such an effective training approach could also result in under-fitting issues. Sun et al. (2023a) find that given the same set of training instructions, LLMs with LoRA perform worse than the fully fine-tuned ones. Furthermore, they also show that when using LoRA, it is preferable to use larger LLMs than larger training instruction datasets because the former solution uses less training costs and achieves better performance than the later one.

4 Alignment Evaluation

After collecting instructions and training LLMs on these instructions, we finally consider the evaluation for alignment quality. In this section, we will discuss benchmarks used for evaluation in Section 4.1 and the evaluation protocols in Section 4.2.

4.1 Evaluation Benchmarks

There are various benchmarks to evaluate the aligned LLMs. In general, these benchmarks can be categorized into *Closed-set Benchmarks* and *Open-set Benchmarks*. The former type focuses on evaluating the skills and knowledge of aligned LLMs, while the latter type often concentrates on the open scenarios where there are no standardized answers.

[7] https://github.com/tloen/alpaca-lora

4.1.1 Closed-set Benchmarks

The closed-set benchmarks mostly include testing instances whose possible answers are predefined and limited to a finite set (e.g., multiple choices). We discuss some of the most commonly used benchmarks below. We refer readers to Chang et al. (2023) for more comprehensive introduction of LLMs' evaluation benchmarks.

General Knowledge MMLU (Hendrycks et al., 2021) is an English-based benchmark to evaluate LLMs knowledge in zero-shot and few-shot settings. It comprehensively includes questions from the elementary level to an advanced professional level from 57 subjects including STEM, the humanities, the social sciences, etc. The granularity and breadth of the subjects make MMLU ideal for identifying LLMs' blind spots. There are also several benchmarks attempting in evaluating the general knowledge in Chinese LLMs. C-MMLU (Li et al., 2023c), C-Eval (Huang et al., 2023), M3KE (Liu et al., 2023a) and AGIEval (Zhong et al., 2023) are all Chinese counterparts of MMLU that include diverse sets of questions from multiple subjects with different difficulty levels from various Chinese standardized exams, including Chinese college entrance exams, advanced maths competitions and law exams. The KoLA benchmark (Yu et al., 2023a) is proposed to evaluate the general real-world knowledge of LLMs.

Reasoning Reasoning is a fundamental type of human intelligence that are crucial in solving complicated tasks. Interestingly, research find that LLMs have exhibit emergent behaviors, including the reasoning ability, when they are sufficiently large. Thus, there are several benchmarks in evaluating the ability of arithmetic, commonsense, and symbolic reasoning for LLMs. GSM8K (Cobbe et al., 2021) and Maths (Hendrycks et al., 2021) are designed to evaluate the arithmetic reasoning ability for LLMs. CSQA (Talmor et al., 2019) and StrategyQA (Geva et al., 2021) are proposed to evaluate the commonsense reasoning ability which requires the LLMs to use daily life commonsense to infer in novel situations. Wei et al. (2022b) propose two novel tasks, Last Letter Concatenation and Coin Flip and measure the Symbolic reasoning ability that involves the manipulation of symbols according to formal rules. BBH (Suzgun et al., 2022), a challenging subset of BIG-Bench (bench authors, 2023), focus on evaluating a wide range

of reasoning skills, such as Date Understanding, Word Sorting, and Causal Judgement.

Coding HumanEval (Chen et al., 2021), HumanEval+ (Liu et al., 2023c), and MBPP (Austin et al., 2021) are extensively used benchmarks to evaluate the coding skills of LLMs. They encompass a vast collection of Python programming problems and corresponding test cases to automatically verify the code generated by Code LLMs. The DS-1000 benchmark (Lai et al., 2022) comprises 1,000 distinct data science workflows spanning seven libraries. It assesses the performance of code generations against test cases and supports two evaluation modes: completion and insertion.

4.1.2 Open-ended Benchmarks

In contrast to the closed-set benchmarks, the responses to open-set benchmarks can be more flexible and diverse, where aligned LLMs are usually given chatting questions or topics that do not have any fixed reference answers. Early attempts of open-ended benchmarks, such as Vicuna-80 (Chiang et al., 2023), Open-Assistant-953 (Kopf et al., 2023), User-Instructions-252 (Wang et al., 2022a), often leverage a small number of syntactic instructions from LLMs as testing instances. All evaluation candidate LLMs are prompted with the same instructions to provide responses, which are then evaluated against human-based or LLMs-based evaluators. However, these types of benchmarks can only provide comparison several LLMs at a time, making it challenging to reveal a fair comparison among a board range of LLMs, as well as incremental updates when new LLMs become available. AlpacaEval (Dubois et al., 2023) tackles this issue by reporting the *Win Rate* of the LLMs candidate to the reference LLM *text-davinci-003*. Accordingly, LLMs with higher *Win Rate* are generally better than the ones with lower *Win Rate*. MT-Bench (Zheng et al., 2023) further increases the evaluation difficulty by proposing 80 multi-turn evaluation instances and wishes LLMs could effectively capture context information in previous turns. FLASK (Ye et al., 2023b) proposed to provide fine-grained evaluation towards aligned LLMs. FLASK includes 1,700 instances from 120 datasets. Each testing instance is labelled with a set of 12 foundational and essential "alignment skills" (e.g., logical thinking, user alignment, etc.). Accordingly, it is straightforward to evaluate LLMs' capabilities on these skills separately.

4.2 Evaluation Paradigm

As open-ended benchmarks often do not have reference answers, it is essential to rely on external human or LLMs evaluators. In this section, we will introduce both human- and LLMs-based evaluation paradigm.

4.2.1 Human-based Evaluation

Automatic metrics, such as BLUE (Papineni et al., 2002) and ROUGE (Lin, 2004), require ground-truth references and have relatively low correlation with human judgments. Thus, they are not feasible for evaluating responses to open-ended questions. To bridge this gap, human annotators are used to evaluate the quality of open-ended model responses. Wang et al. (2022a); Wu et al. (2023) propose to evaluate the response quality in an ordinal classification setting where human annotators are instructed to categorize each response into one of the four levels (i.e., acceptable, minor errors, major errors and unacceptable), separately. However, some other research have found that such classification annotation strategy heavily depend on the subjectivity of annotators, which can result in poor inter-rater reliability (Kalpathy-Cramer et al., 2016). Accordingly Taori et al. (2023) propose to use a pairwise comparison framework for evaluating the output quality of two LLMs systems. Given the instruction inputs and two model outputs, the human annotators are asked to select a better one. Furthermore, to accurately evaluate multiple LLMs, Zheng et al. (2023); Dettmers et al. (2023) further introduce the Elo rating system which calculates the relative skill levels of players in zero-sum games such as chess games. Specifically, in Elo system, the player scores are updated based on the result of each pairwise comparison and the current player scores.

4.2.2 LLMs-based Evaluation

While human evaluations are often of high quality, it could be inefficient and expensive. In addition, the increasing quality of generated text from LLMs makes it more challenging for human annotators to distinguish between human-written and LLM-generated text in the open-ended NLP tasks (Clark et al., 2021). Given the strong text capability of LLMs, recent studies propose to incorporate LLMs into the output text evaluation in various NLP tasks without additional expensive references and human efforts. Tang et al. (2023) propose to improve the traditional automatic metrics by increasing the

number of references via LLMs-based paraphrasing systems. However, such method still requires one reference for each evaluation instance. In contrast, Liu et al. (2023e); Fu et al. (2023); Chen et al. (2023d); Chiang and Lee (2023) propose to directly use LLMs to evaluate the generated text quality without a single reference in a wide range of Natural Language Generation (NLG) tasks. Specifically, they construct complicated input instructions with tasks background and evaluation rules and prompt LLMs to follow these evaluation instructions to provide scores for output text. There are also some research efforts that propose LLMs-based evaluation framework for specific NLG tasks, including text summarization Gao et al. (2023), code generation (Zhuo, 2023), open-ended QA (Bai et al., 2023) and conversations (Lin and Chen, 2023). Due to the flexibility of prompts, it is also possible to conduct multi-dimensional evaluation towards the generated text (Lin and Chen, 2023; Fu et al., 2023). Min et al. (2023); Zha et al. (2023) propose to evaluate factual correctness using both closed-sourced and open-sourced LLMs. Similar to human evaluation, there are also research efforts in explicitly prompting LLMs to conduct pairwise comparisons. To compare the capabilities of two LLMs, instead of assigning scores separately, Dubois et al. (2023); Zheng et al. (2023) explicitly to prompt GPT-4 to select the better response for the same instruction inputs.

LLMs Evaluation Bias Despite LLMs achieve impressive consistency with human judgment, Wang et al. (2023b) find that such LLM-based evaluation paradigm suffers from a positional bias and those strong LLMs (i.e., GPT-4) tend to assign higher scores to the first appeared candidates. To calibrate such bias, they propose to **a)** repeat the LLM evaluation process multiple times with different candidate ordering and **b)** explicitly prompt LLMs to provide chain-of-thoughts for the evaluation before assigning the actual score. (Wu and Aji, 2023) find that LLM-based evaluation prefer candidates with factual errors over shorter candidates and candidates with grammatical errors, despite the former one could impose greater danger than the latter ones. To address this bias, they propose a multi-dimensional Elo rating system which separately evaluates the candidates from the perspective of accuracy, helpfulness and language. Such approach allows a more comprehensive understanding towards the candidates quality than previous one-shot evaluation. Concretely, (Zheng et al., 2023) systematically show the bias LLMs-based evaluation systems. On top of positional and length bias, they also discover Self-enhancement bias which means LLMs favor their own responses than the ones from other sources. To tackle these biases, their solutions include swapping responses, adding few-shot examples and leveraging CoT and references information.

Evaluation-Specific LLM Despite achieving high-quality automatic evaluation results, the above approaches heavily rely on state-of-the-art closed-source LLMs (e.g., GPT-4) which could result in data privacy issues. (Zheng et al., 2023) propose to train evaluation-specific LLMs. PandaLM (Wang et al., 2023c) is such a specialized evaluation LLMs by fine-tuning LLaMA-7B using around 300K high-quality synthetic evaluation instructions generated from GPT-3.5. Specifically, they first collect large volumes of instructions as well as outputs from a diverse range of open-sourced LLMs, such as LLaMA-7B and Bloom-7B. They then prompt GPT-3.5 to analysis and evaluate the quality of a pair of outputs. Their results on human-annotated meta-evaluation shows that, despite bebing much smaller, PandaLM achieves on-par evaluation performance comparing to GPT-3.5 and GPT-4.

5 Challenges and Future Directions

The development of LLM alignment is still in a rudimentary stage and thus leaves much room for improvement. In this section, we summarize existing important research efforts of aligning LLMs with human in Table 1. Below, we will discuss some of the challenges as well as the corresponding future research directions.

Fine-grained Instruction Data Management While research on LLMs alignment have been unprecedentedly active, many of these research efforts propose to leverage training instructions from diverse sources, making it challenging to fairly compare among different methods. As discussed in Section 2.3, there are some interesting findings about the implication of particular instruction dataset. For example, FLAN and programming instructions can improve reasoning capability aligned LLMs (Ghosal et al., 2023) and ShareGPT general performs well across a wide range of benchmarks (Wang et al., 2023d). However, there are still many issues in other aspects of instruction

Aligned LLM	Size	Lang.	Initial LLMs	Training	Self Instruction	NLP Benchmarks	Human Annotations	Human Eval	Auto. Benchmark Eval	LLM Eval
Alpaca (Taori et al., 2023)	7B	EN	LLaMA	SFT	Text-Davinci-003	✗	✗	Author Verification	✗	✗
Vicuna (Chiang et al., 2023)	7B, 13B, 33B	EN	LLaMA	SFT	GPT-3.5	✗	70K ShareGPT	✗	✗	*Vicuna-80*
GPT4ALL (Anand et al., 2023)	6B, 13B	EN	LLaMA GPT-J	SFT	✗	Bloomz-P3	OIG, ShareGPT, Dolly Stack Overflow	✗	Common Sense Reasoning	✗
LLaMA-GPT4 (Peng et al., 2023)	7B	EN, CN	LLaMA	SFT	Text-Davinci-003 GPT-4	✗	✗	*User-Instructions-252* Pairwise, AMT	Unnatural Instructions	*Vicuna-80*
Phoenix (Chen et al., 2023e)	7B, 13B	Multilingual	LLaMA BLOOMZ	SFT	GPT-3.5 Multilingual and Dialogue Data	✗	ShareGPT	Volunteers	✗	GPT-3.5, GPT-4
UltraLLaMA (Ding et al., 2023)	13B	EN	LLaMA	SFT	GPT-3.5 Dialogue Data	✗	✗	✗	Truthful QA	GPT 3.5 *Vicuna-80* 300 diverse questions
Baize (Xu et al., 2023c)	7B, 13B, 30B	EN	LLaMA	Revision, LoRA	GPT-3.5 self-Chat Data	✗	Quora Questions	✗	✗	GPT-4
WizardLM (Xu et al., 2023b)	7B, 13B, 30B	EN	LLaMA	SFT	GPT-3.5, Alpaca Complex Instructions	✗	ShareGPT	10 Annotators Pairwise Comparison	✗	GPT-4, *WizedLM-218*
WizardCoder (Luo et al., 2023)	15B	EN, Code	StarCoder	SFT	GPT-3.5, Code Alpaca Complex Instructions	✗	✗	✗	HumanEval, MBPP HumanEval+, DS-1000	✗
OpenChat (Wang et al., 2023a)	13B	EN	LLaMA	Language	✗	✗	GPT 3.5 & GPT4 ShareGPT	✗	MMLU	GPT-4
Guanaco (Dettmers et al., 2023)	13B, 33B, 65B	EN	LLaMA	QLoRA	Alpaca, SELF-INSTRUCT Unnatural instructions GPTeacher, Guanaco Baize Instructions	FLAN	Chip2	Elo, *Vicuna-80*	MMLU	Elo, *Vicuna-80* *Open-Assistant-953*
MPT-chat (Team, 2023)	13B, 30B	EN	MPT	SFT	✗	Vicuna ShareGPT	✗	✗	MMLU	GPT4, MT-bench
FLACUNA (Ghosal et al., 2023)	13B	EN	Vicuna	LoRA	Alpaca, Code Alpaca	FLAN	ShareGPT	✗	MMLU, BBH, DROP CRASS, HumanEval	GPT 3.5, IMPACT
Bactrian-X (Li et al., 2023b)	7B	Multilingual	LLaMA BLOOMZ	LoRA	Alpaca Google Translation	✗	✗	✗	XCOPA, XStoryCloze XWinograd, SentimentX	GPT 4 Multilingual *Vicuna-80*
Ocra (Mukherjee et al., 2023)	13B	EN	LLaMA	SFT	✗	FLAN	✗	✗	AGIEval, BBH	GPT-4, *Vicuna-80* *WizedLM-218*, *Awesome-164*
Phi-1 (Gunasekar et al., 2023)	350M, 1.3B	EN, Code	Phi-1-base	SFT	GPT-3.5 Synthetic Textbook	✗	Python, The Stack Stack Overflow	✗	HumanEval	GPT-4 Grading
Chinese Alpaca (Cui et al., 2023b)	7B, 13B, 33B	EN, CN	Chinese LLaMA	LoRA	STEM Org. and Trans. Alpaca	pCLUE	✗	✗	C-Eval	✗
Lion (Jiang et al., 2023)	7B, 13B	EN	LLaMA	SFT	Alpaca GPT 3.5 Adv. Instruction	✗	✗	HHH *User-Instructions-252*	✗	GPT-4, *Vicuna-80*
Stable Alignment (Liu et al., 2023d)	7B	EN	Alpaca	SFT	GPT-3.5 Social Aligned Instructions	✗	✗	✗	✗	GPT-4 HHH, HHH-A
Dromedary (Sun et al., 2023b)	65B	EN	LLaMA	SFT	LLaMA-65B, Self-Align	✗	175 Munnal Examples 16 Principle Rules	✗	TruthfulQA, BBH	GPT-4, *Vicuna-80*
Dolly-v2 (Conover et al., 2023)	3B, 7B, 12B	EN	Pythia	SFT	✗	✗	*databricks-dolly-15k*	✗	*LLM Harness*	✗
Selfee (Ye et al., 2023a)	7B, 13B	EN	LLaMA	Revision	GPT 3.5 Self-Improve Alpaca	FLAN, Maths, Code	ShareGPT	✗	✗	GPT-4, *Vicuna-80*
TÜLU (Wang et al., 2023d)	7B, 13B, 30B, 65B	EN	LLaMA	SFT	Alpaca, Code Alpaca GPT4-Alpaca, Self-instruct	FLAN, CoT	Dolly, ShareGPT Open Assistant	Acceptability Pairwise Comparison	MMLU, GSM, BBH TydiQA, Codex-Eval	GPT4 on *Vicuna-80*, Koala Open Assistant Benchmarks
Koala (Geng et al., 2023)	13B	EN	LLaMA	Language	Alpaca	✗	OIG, HC3, Anthropic HH OpenAI WebGPT, Summary	100 AMT Annotators on *Alpaca and Koala Test*	✗	✗
Bayling (Zhang et al., 2023c)	7B, 13B	Multilingual	LLaMA	SFT	Alpaca GPT 3.5 Interactive Translation	✗	ShareGPT	Translation Quality	WMT22 Multilingual Translation Lexically Constrained Translation	✗
Wombat (Yuan et al., 2023)	7B	EN	Alpaca	Rank	Alpaca ChatGPT Ratings	✗	*Helpful and Harmless*	✗	✗	GPT-4, *Vicuna-80*
Lamini-lm (Wu et al., 2023)	0.7B	EN	T5-Flan	SFT	Alpaca Self-instruct	P3, FLAN	✗	Human Rating	*LLM harness*	✗

Table 1: An overview of popular aligned LLMs, including their Size, supported languages, initial LLMs, alignment training method, alignment data, and alignment evaluation.

data management remaining unclear, including the optimal quality control towards instruction data, optimal instruction training sequence, how to effectively mix-up different instructions. These research efforts could finally enable fine-grained instruction management, allowing researchers and practitioners to construct high-quality instruction data.

LLMs Alignment for non-English Languages Most of existing research in LLMs alignment are English-dominated. While many approaches, such as complex instruction generation (Xu et al., 2023b) and explanation tuning (Mukherjee et al., 2023), are language-agnostic, they only explore English-based prompts and it is unclear how well these prompts perform when adapting to other languages, severely hindering the application of LLMs to non-English regions. It is interesting to see *1)* how these alignment technologies perform in various languages, in particular low-resource languages, and *2)* how to effectively transfer the effect of LLMs alignment across different languages.

LLMs Alignment Training Technologies As shown in Table 1, most of existing aligned LLMs are based on the simple SFT technology. However, SFT does not explicitly incorporate human preference into LLMs. As a result, aligning LLMs solely based on SFT could require a lot more instruction data and training resources. In general, there is a lacking of comprehensive investigation over the effect of various training technologies to incorporate human preference into LLMs. Thus, it is critical to come up with resource-constrained LLM alignment training framework where certain alignment resources are given at a certain level (e.g., maximum 10K instructions, 5 hours training time, etc.), allowing researchers and practitioners to verify the effectiveness of various training methods. As increasing number of instruction data have become available, this exploration could further promote effective and environmental-friendly LLMs alignment solutions.

Human-in-the-loop LLMs Alignment Data Generation Table 1 has shown that ShareGPT data has been widely adapted for LLMs alignment. The preliminary analysis in Wang et al. (2023d) also reveal that ShareGPT performs consistly well across a wide range of NLP tasks. These results indicate that human is still a key factor in improving LLMs alignment quality. Different from traditional human annotation framework where human provides annotation based on the instructions, ShareGPT is a human-in-the-loop alignment solution where human can freely determine what LLMs should generate. This shows the great potential of human-in-the-loop data generation solution in LLMs alignment. It will be interesting to explore other types of human-in-the-loop solutions to further facilitate LLMs alignment.

Human-LLM Joint Evaluation Framework Existing LLM evaluation frameworks either use LLMs for effective evaluation or leverage crowdsourcing for high-quality evaluation. As shown in (Wu and Aji, 2023; Liu et al., 2023e), state-of-the-art LLMs have demonstrated similar or superior evaluation capability in various NLP tasks. It is feasible to use LLMs as special evaluation annotators and develop LLM-human joint evaluation framework where LLMs and human are assigned with different evaluation tasks based on their own strengths to maintain both efficiency and quality of the evaluation procedure for LLM alignment .

6 Conclusion

This survey provides an up-to-date review to recent advances of LLMs alignment technologies. We summarize these research efforts into *Alignment Instruction Collection*, *Alignment Training* and *Alignment Evaluation*. Finally, we pointed out several promising future directions for LLMs alignment. We hope this survey could provide insightful perspectives and inspire further research in improving LLMs alignment.

References

Waseem AlShikh, Manhal Daaboul, Kirk Goddard, Brock Imel, Kiran Kamble, Parikshith Kulkarni, and Melisa Russak. 2023. Becoming self-instruct: introducing early stopping criteria for minimal instruct tuning. *arXiv preprint arXiv:2307.03692*.

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. https://github.com/nomic-ai/gpt4all.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Stephen Bach, 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 Alshaibani, 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, Dublin, Ireland. Association for Computational Linguistics.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. 2023. Benchmarking foundation models with language-model-as-an-examiner. *arXiv preprint arXiv:2306.04181*.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. *arXiv preprint arXiv:2307.06290*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Jiaao Chen, Aston Zhang, Xingjian Shi, Mu Li, Alex Smola, and Diyi Yang. 2023a. Parameter-efficient fine-tuning design spaces. In *The Eleventh International Conference on Learning Representations*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023b. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023c. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023d. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. *arXiv preprint arXiv:2304.00723*.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2023e. Phoenix: Democratizing chatgpt across languages. *CoRR*, abs/2304.10453.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Annual Meeting of the Association for Computational Linguistics*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Yiming Cui, Ziqing Yang, and Xin Yao. 2023a. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023b. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

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

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

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Yann Dubois, 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. *arXiv preprint arXiv:2305.14387*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. Blog post.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Deepanway Ghosal, Yew Ken Chia, Navonil Majumder, and Soujanya Poria. 2023. Flacuna: Unleashing the problem solving power of vicuna using flan fine-tuning. *arXiv preprint arXiv:2307.02053*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

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. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *CoRR*, abs/2212.09689.

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

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

Sophie Jentzsch and Kristian Kersting. 2023. Chatgpt is fun, but it is not funny! humor is still challenging large language models. *arXiv preprint arXiv:2306.04563*.

Yunjie Ji, Yan Gong, Yong Deng, Yiping Peng, Qiang Niu, Baochang Ma, and Xiangang Li. 2023. Towards better instruction following language models for chinese: Investigating the impact of training data and evaluation. *CoRR*, abs/2304.07854.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of closed-source large language model. *ArXiv*, abs/2305.12870.

Jayashree Kalpathy-Cramer, J. Peter Campbell, Deniz Erdogmus, Peng Tian, Dharanish Kedarisetti, Chace Moleta, James D. Reynolds, Kelly Hutcheson, Michael J. Shapiro, Michael X. Repka, Philip Ferrone, Kimberly Drenser, Jason Horowitz, Kemal Sonmez, Ryan Swan, Susan Ostmo, Karyn E. Jonas, R.V. Paul Chan, Michael F. Chiang, Michael F. Chiang, Susan Ostmo, Kemal Sonmez, J. Peter Campbell, R.V. Paul Chan, Karyn Jonas, Jason Horowitz, Osode Coki, Cheryl-Ann Eccles, Leora Sarna, Audina Berrocal, Catherin Negron, Kimberly Denser, Kristi Cumming, Tammy Osentoski, Tammy Check, Mary Zajechowski, Thomas Lee, Evan Kruger, Kathryn McGovern, Charles Simmons, Raghu Murthy, Sharon Galvis, Jerome Rotter, Ida Chen, Xiaohui Li, Kent Taylor, Kaye Roll, Jayashree Kalpathy-Cramer, Deniz Erdogmus, Maria Ana Martinez-Castellanos, Samantha Salinas-Longoria, Rafael Romero, Andrea Arriola, Francisco Olguin-Manriquez, Miroslava Meraz-Gutierrez, Carlos M. Dulanto-Reinoso, and Cristina Montero-Mendoza. 2016. Plus disease in retinopathy of prematurity: Improving diagnosis by ranking disease severity and using quantitative image analysis. *Ophthalmology*, 123(11):2345–2351.

Andreas Kopf, Yannic Kilcher, Dimitri von Rutte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Rich'ard Nagyfi, ES Shahul, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. *ArXiv*, abs/2304.07327.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Scott Wen tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2022. Ds-1000: A natural and reliable benchmark for data science code generation. *ArXiv*, abs/2211.11501.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: communicative agents for "mind" exploration of large scale language model society. *CoRR*, abs/2303.17760.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023b. Bactrian-x: A multilingual replicable instruction-following model with low-rank adaptation. *arXiv preprint arXiv:2305.15011*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023c. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

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, Online. Association for Computational Linguistics.

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

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *arXiv preprint arXiv:2305.13711*.

Chuang Liu, Renren Jin, Yuqi Ren, Linhao Yu, Tianyu Dong, Xiaohan Peng, Shuting Zhang, Jianxiang Peng, Peiyi Zhang, Qingqing Lyu, et al. 2023a. M3ke: A massive multi-level multi-subject knowledge evaluation benchmark for chinese large language models. *arXiv preprint arXiv:2305.10263*.

Hao Liu, Xinyang Geng, Lisa Lee, Igor Mordatch, Sergey Levine, Sharan Narang, and P. Abbeel. 2022a. Towards better few-shot and finetuning performance with forgetful causal language models.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023b. Languages are rewards: Hindsight finetuning using human feedback. *arXiv preprint arXiv:2302.02676*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023c. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *arXiv preprint arXiv:2305.01210*.

Ruibo Liu, Chenyan Jia, Ge Zhang, Ziyu Zhuang, Tony X Liu, and Soroush Vosoughi. 2022b. Second thoughts are best: Learning to re-align with human values from text edits. In *Advances in Neural Information Processing Systems*.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. 2023d. Training socially aligned language models in simulated human society. *arXiv preprint arXiv:2305.16960*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023e. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022c. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

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

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. 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, Dublin, Ireland. Association for Computational Linguistics.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Huu Nguyen, Sameer Suri, Ken Tsui, and Christoph Schuhmann. 2023. The oig dataset.

Tung Nguyen, Qinqing Zheng, and Aditya Grover. 2022. Conserweightive behavioral cloning for reliable offline reinforcement learning. *arXiv preprint arXiv:2210.05158*.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*.

Xianghui Sun, Yunjie Ji, Baochang Ma, and Xiangang Li. 2023a. A comparative study between full-parameter and lora-based fine-tuning on chinese instruction data for instruction following large language model. *arXiv preprint arXiv:2304.08109*.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David D. Cox, Yiming Yang, and Chuang Gan. 2023b. Principle-driven self-alignment of language models from scratch with minimal human supervision.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench

tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyi Tang, Hongyuan Lu, Yuchen Eleanor Jiang, Haoyang Huang, Dongdong Zhang, Wayne Xin Zhao, and Furu Wei. 2023. Not all metrics are guilty: Improving nlg evaluation with llm paraphrasing. *arXiv preprint arXiv:2305.15067*.

Rohan Taori, 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`.

MosaicML NLP Team. 2023. Introducing mpt-30b: Raising the bar for open-source foundation models. Accessed: 2023-06-22.

Guan Wang, Sijie Cheng, Qiying Yu, and Changling Liu. 2023a. OpenChat: Advancing Open-source Language Models with Imperfect Data.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023c. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023d. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*.

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, 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*.

Minghao Wu and Alham Fikri Aji. 2023. Style over substance: Evaluation biases for large language models. *ArXiv*, abs/2307.03025.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *CoRR*, abs/2304.14402.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023a. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*.

Can Xu, 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.

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

Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. 2023a. Selfee: Iterative self-revising llm empowered by self-feedback generation. Blog post.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023b. Flask: Fine-grained language model evaluation based on alignment skill sets.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. 2023a. Kola: Carefully benchmarking world knowledge of large language models. *arXiv preprint arXiv:2306.09296*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023b. Large language model as attributed training data generator: A tale of diversity and bias. *arXiv preprint arXiv:2306.15895*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

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

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023b. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023c. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *ArXiv*, abs/2306.10968.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. 2023. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Terry Yue Zhuo. 2023. Large language models are state-of-the-art evaluators of code generation. *arXiv preprint arXiv:2304.14317*.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Appendix

Table 2: The outputs of original LLaMA and Chinese Tokenizer. This example is from Cui et al. (2023b).

Inputs: 人工智能是计算机科学、心理学、哲学等学科融合的交叉学科。
LLaMA: _, 人, 工, 智, 能, 是, 计, 算, 机, 科, 学, 、, 心, 理, 学, 、, 0xE5, 0x93, 0xB2, 学, 等, 学, 科, 0xE8, 0x9E, 0x8D, 合, 的, 交, 0xE5, 0x8F, 0x89, 学, 科, 。
Chinese: _, 人工智能, 是, 计算机, 科学, 、, 心理学, 、, 哲学, 等, 学科, 融合, 的, 交叉, 学科, 。

A.1 Training Language-Specific LLMs

Existing LLMs described above are mostly English-oriented. Thus, it becomes necessary to adapt the superior linguistic ability to other languages. Ji et al. (2023); Cui et al. (2023b) demonstrate existing English-dominated LLaMA has less than 1,000 Chinese characters in its vocabulary and LLaMA has to represent Chinese characters using the byte-based fallback strategy, which significantly increases input length and decreases the inference efficiency. As shown in Table 2, compared to the default LLaMA tokenizer, the specialized Chinese tokenizer trained using large-scale Chinese corpus can produce more compact and semantically meaningful token representations (e.g., long and complex Chinese phrases). To leverage the linguistic knowledge in orginal LLaMA, Cui et al. (2023b) propose a two-stage Chinese pre-training solution to enable LLaMA to better understand Chinese inputs. Before training they first add 20K Chinese words and phrases into the existing LLaMA vocabulary. In the first stage, they only train the input word embeddings and keep the rest parameters in LLaMA frozen. In the second stage, to save training resources, they add LoRA parameters and jointly train the parameters in the input word embeddings, self-attentive heads and LoRA parameters. Ji et al. (2023) also report the benefits of such strategy under a GPT-4 evaluation framework.