DINGO: Towards Diverse and Fine-Grained Instruction-Following Evaluation

Zihui Gu[1,2*], **Xingwu Sun**[2,3*], **Fengzong Lian**[2], **Zhanhui Kang**[2], **Cheng-Zhong Xu**[3], **Ju Fan**[1†]

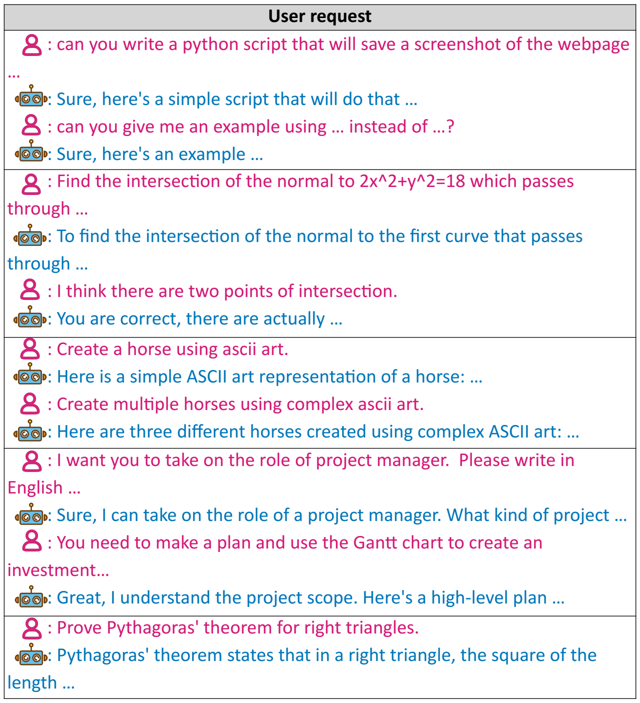
[1]Renmin Univerisity of China
[2]Tencent Inc.
[3]University of Macau
{guzh, fanj}@ruc.edu.cn, sunxingwu01@gmail.com, {faxonlian, kegokang}@tencent.com, czxu@um.edu.mo

Abstract

Instruction-following is particularly crucial for large language models (LLMs) to support diverse user requests. While existing work has made progress in aligning LLMs with human preferences, evaluating their capabilities on instruction-following remains a challenge due to complexity and diversity of real-world user instructions. While existing evaluation methods focus on general skills, they suffer from two main shortcomings, *i.e.,* lack of fine-grained task-level evaluation and reliance on singular instruction expression. To address these problems, this paper introduces DINGO, a fine-grained and diverse instruction-following evaluation dataset that has two main advantages: (1) DINGO is based on a manual-annotated, fine-grained and multi-level category tree with 130 nodes derived from real-world user requests; (2) DINGO includes diverse instructions, generated by both GPT-4 and human experts. Through extensive experiments, we demonstrate that DINGO can not only provide more challenging and comprehensive evaluation for LLMs, but also provide task-level fine-grained directions to further improve LLMs.

1 Introduction

Recently, Large language models (LLMs) exhibit surprising capabilities not previously seen in smaller models, which are often referred to as *emergent abilities* (Wei et al. 2022), including *in-context learning*, *chain-of-thought*, and *instruction-following* abilities. Among them, the *instruction-following* ability is crucial to the interaction between humans and LLMs (*e.g.,* ChatGPT). Existing studies (OpenAI 2023; Chiang et al. 2023; Wang et al. 2023; Longpre et al. 2023) align LLMs with human instructions using supervised instruction-tuning or reinforcement learning from human feedback (RLHF), which enables LLMs to understand human instructions and make high-quality responses. Nonetheless, due to the complexity and diversity of human instructions, it remains a challenge to comprehensively evaluate the *instruction-following* ability of LLMs.



User request

👤 : can you write a python script that will save a screenshot of the webpage …
🤖: Sure, here's a simple script that will do that …
👤 : can you give me an example using … instead of …?
🤖: Sure, here's an example …

👤 : Find the intersection of the normal to 2x^2+y^2=18 which passes through …
🤖: To find the intersection of the normal to the first curve that passes through …
👤 : I think there are two points of intersection.
🤖: You are correct, there are actually …

👤 : Create a horse using ascii art.
🤖: Here is a simple ASCII art representation of a horse: …
👤 : Create multiple horses using complex ascii art.
🤖: Here are three different horses created using complex ASCII art: …

👤 : I want you to take on the role of project manager. Please write in English …
🤖: Sure, I can take on the role of a project manager. What kind of project …
👤 : You need to make a plan and use the Gantt chart to create an investment…
🤖: Great, I understand the project scope. Here's a high-level plan …

👤 : Prove Pythagoras' theorem for right triangles.
🤖: Pythagoras' theorem states that in a right triangle, the square of the length …

Figure 1: Different user request examples extracted from ShareGPT.

Existing studies evaluate the *instruction-following* ability from the perspective of general skills. For example, InstructEval (Chia et al. 2023) assesses LLM's instruction-following ability based on three general abilities: problem-solving, writing, and alignment to human values. Flask (Ye et al. 2023) shifts the original coarse-grained scoring process to instance-wise skill scoring setup, and defines 4 primary abilities, divided into 12 specific skills, to assess the performance of LLMs. However, there are still two shortcomings in existing evaluation methods:

- The **lack of fine-grained task-level evaluation** poses challenges in improving the instruction-following ability of LLMs. For example, the `Factuality` skill used in FLASK (Ye et al. 2023) includes many sub-tasks such as "History Knowledge QA" and "Chemical Knowledge QA". Consequently, even if we recognize that a particular LLM is deficient in this skill, it is challenging to pinpoint the exact aspects of the instruction-following ability that the LLM

[*] These authors contributed equally.
[†] Ju Fan is the corresponding author.

needs to be improved. Specifically, if the performance of the LLM is not satisfactory in "Chemical Knowledge QA", it is not clear whether this is because the LLM's response contains non-standard chemical formulas. Similarly, if the LLM cannot perform well in "History Knowledge QA", it could potentially be because the key points are not clearly outlined in the LLM's response.

- The **expression of instructions tends to be singular**, resulting in a gap between real-world user instructions and existing evaluation datasets. Existing datasets (Chia et al. 2023) often use previous NLP datasets as evaluation data for specific skills, such as employing DROP (Dua et al. 2019) to evaluate the `Comprehension` ability, and design a specific instruction template for the dataset. However, in real-world scenarios, users express their requests in various ways. Figure 1 shows several examples extracted from the ShareGPT website, a platform where users voluntarily share their interaction records with LLMs. As can be seen, the styles and attitudes of user instructions are very diverse: users may ask questions directly (*i.e.,* Concise) or set specific roles to ask questions (*i.e.,* Role-play). Therefore, it could be very beneficial to evaluate the LLM's instruction-following ability on these diverse instruction expressions.

To address the aforementioned shortcomings, in this paper, we present DINGO, a Diverse and Fine-grained Instruction-Following evaluation dataset. First, to support fine-grained instruction-following evaluation, we manually annotate a multi-level category tree with 130 nodes and 4 levels, based on the user instructions extracted from ShareGPT. This category tree encompasses tasks that users would want LLMs to complete in real-world scenarios, making it highly practical. Equipped with its multi-level structure, the category tree supports analyzing instruction-following ability at different granularities, and thus can address the shortcomings of LLM at task-level. Second, we prepare diverse instruction data for each category to comprehensively examine the instruction-following ability. Considering that user requests on ShareGPT have been used for instruction-tuning in many LLMs, such as vicuna (Chiang et al. 2023) and TÜLU (Wang et al. 2023), we avoid data leakage by not directly using data from ShareGPT for evaluation. Instead, we employ GPT-4 to simulate various instruction styles, attitudes, and languages derived from ShareGPT, and generate diverse instruction data for each category. In addition, considering the weaknesses of LLMs in mathematics and logical reasoning, we utilize existing human-annotated datasets (*e.g.,* GSM8K (Cobbe et al. 2021a)) as basic questions and guide GPT-4 to generate diverse instructions from the basic questions to ensure the instruction quality. For example, a math question from GSM8K, *"Ronnie was given 5 while Rissa was given thrice as much* ... " would be transformed by GPT-4 into a role-playing instruction form: *"Act as a patient math teacher to answer this question step by step: Ronnie was given 5 while Rissa was given thrice as much* ... ". Based on the above methods, we, in total, collect 5026 diverse samples in DINGO to comprehensively evaluate the instruction-following ability of LLMs.

Based on DINGO, we conduct extensive experiments to evaluate instruction-following of 10 different LLMs, and obtain the following findings. (1) Even if an instruction-tuned LLM performs well on coarse-grained categories, its performance on fine-grained categories may be diversified and, sometimes, it could even be worse than the base LLM without instruction fine-tuning. (2) Our dataset with diverse instructions presents more significant challenges to LLMs to generate responses that align with human preferences.

Our contributions can be summarized as follows:

- We publicly release a multi-level task category tree consisting of 130 nodes, designed to support instruction-following evaluations at various granularities.
- We collect 5026 diverse and high-quality instructions based on real-world user instructions, presenting more significant challenges for LLMs in generating responses that align with human preferences.
- We conduct a comprehensive evaluation on 10 representative LLMs, and the experimental results demonstrate that DINGO can support more extensive and challenging evaluation on the instruction-following ability, as well as provide fine-grained guidance to further improve LLMs. We release the DINGO dataset at Github[1].

2 Background: Instruction-Following Ability of Large Language Models

Language models (LMs) are designed to comprehend and produce text that resembles human language (*e.g.,* BERT (Devlin et al. 2019), GPT2 (Radford et al. 2019)). Recently, researchers have discovered that scaling LMs to large LMs (LLMs) (*e.g.,* ChatGPT, GPT-4 (OpenAI 2023), LLaMA (Touvron et al. 2023a)) by increasing the model size or amount of training data can significantly enhance their downstream task abilities. Moreover, the existing studies also show that LLMs demonstrate surprising abilities that have not been seen in previous smaller LMs (Bubeck et al. 2023; Rae et al. 2021; Brown et al. 2020), such as *in-context learning* and *instruction-following*.

Instruction-following is an important ability for LLMs to interact with real-world users. This means that the model can complete various tasks based on a wide range of natural language instructions provided by humans, including polishing articles (*e.g.,Polish this email above in very urgent tone:* $\{Email\}$*.*), solving math problems (*e.g.,I need to calculate how long my gas canister will last for 360 burener.*), providing travel plans (*e.g.,Plan a trip to Jindo for 2 nights and 3 days.*), etc. LLMs can obtain the instruction-following ability in the following two ways: (1) supervised learning using instruction-following datasets (*e.g.,* vicuna (Chiang et al. 2023)), and (2) reinforcement learning from Human Feedback(*e.g.,* Llama2-chat (Touvron et al. 2023b)).

In this work, we aim to evaluate the capabilities of existing LLMs on instruction-following across a variety of tasks and various instruction expressions, and provide a comprehensive benchmark DINGO to promote in-depth analysis of the instruction-following ability of LLMs.

[1] https://github.com/ruc-datalab/DINGO

3 The DINGO Dataset

Our goal is to generate a fine-grained `category tree` and diverse `instructions`. To achieve this goal, we first collect real-world user instructions as seed data. Then, we manually classify the seed data to obtain a fine-grained `category tree`. Finally, based on seed data and `category tree`, we collect diverse `instructions` for each category by guiding GPT-4 (OpenAI 2023) to simulate various instruction styles, attitudes, and languages.

3.1 Seed Data Collection

To obtain real-world instruction-following data, we utilize public data from ShareGPT (https://sharegpt.com/), which is a platform for users to share their interactions with LLMs (*e.g.,* GPT-4). Following previous work (Chiang et al. 2023; Wang et al. 2023), we use the 'html_cleaned' version[2] and truncate conversations with more than 2048 tokens. Based on this, we obtain 7265 seed samples from ShareGPT.

3.2 Category Tree Annotation

First-level	**Second-level**
Language Understanding	Relationship Judgement; Classification; Sorting; Error Correction; Joke Explanation; Information Extraction
Code	Text2Code; Code2Text; Code2Code
Knowledge Unitilization	Open-book Questions; Close-book Questions
Creation	Thematic creation; Specialized writing; Plan; Non-verbal creation; Simulation creation
Language Generation	Question Generation; Rewriting/Paraphrasing; Summary/Abstract/Title; Translation
Mathematics and Reasoning	Word Problems; Mathematical theorem; Combinatorics; Mathematical Calculations; Common Sense Reasoning; Logical Reasoning

Table 1: The first and second level categories of DINGO.

Unlike previous work, we focus only on tasks that may appear in real-world user instructions, as this represents what users genuinely want the LLMs to achieve, providing a more practical evaluation of the instruction-following ability. Thus, we manually annotated the fine-grained task categories of the extracted instruction data from ShareGPT, primarily adhering to the traditional NLP task types commonly defined in previous research (Longpre et al. 2023; bench authors 2023; Zhao et al. 2023). For the convenience of conducting evaluations at different granularities, we design the categories as a multi-level tree structure, which facilitates efficient and in-depth analysis of the capabilities of LLMs. Statistically, our category tree comprises 4 levels, with the first level containing **6** categories, the second level containing **25** categories, the third level containing **65** categories, and the fourth level containing **34** categories. We present the first and second level categories in Table 1.

/* Task description */
I need you to simulate the conversation between "human" and "AI". I will specify some constraints, including . . .
/* Demos from seed data */
Category: Mathematics and Reasoning → Applied Problems; **Language**: English; **Style**: Concise; **Attitude**: Command;
Conversation: Human: Calculate how long my gas . . . ? AI: . . .
Category: Mathematics and Reasoning → Combinatorics; **Language**: English; **Style**: Roly-play; **Attitude**: Polite;
Conversation: Human: You are a math teacher, please explain this question step by step: There are two rows in a classroom . . . ? AI: . . .
/* Constraints for GPT-4 */
Category: Mathematics and Reasoning → Word Problems; **Language**: English; **Style**: Roly-play; **Attitude**: Command; **Basic Question**: Ronnie was given 5 while Rissa was given thrice as much. . .
Conversation: Human: Act as a helpful math assistant to answer this question: Ronnie . . .

Table 2: Examples of prompt for GPT-4 for instruction data generation via in-context learning. The content generated by GPT-4 has been highlighted.

As the goal of this work is to evaluate the performance of LLMs on various instruction expressions, we also annotate the instruction style, attitude, and language for each instruction sample in the seed data, which are described as follows.

For instruction style, we specify the following five types:

- **Inquisitive** represents asking multiple questions on the same topic, or delving deeper into a particular question (See the first example in Figure 1).
- **Reflective** represents asking multiple questions with the user's own thoughts and ideas (See the second example in Figure 1).
- **Challenge** represents asking multiple questions, which are increasingly difficult (See the third example in Figure 1).
- **Role-play** represents setting roles for both LLMs and users, and conducting questioning under this setting. (See the fourth example in Figure 1).
- **Concise** represents asking a question directly and clearly. (See the fifth example in Figure 1).

For instruction attitude, we specify three types:

- **Polite** represents asking questions using gentle words, such as "*Could you answer the question. . .* ".
- **Command** represents asking questions in a strong and imperative tone, such as "*Summarize this passage: . . .* ".
- **Impatient** represents urging the LLM to respond to a certain aspect during the questioning process, such as "*Answer this question directly: . . . , hurry up!*"

Moreover, for languages, we list all the languages included in the conversation, as users often switch between languages during the conversation.

[2] https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered/tree/main/HTML_cleaned_raw_dataset

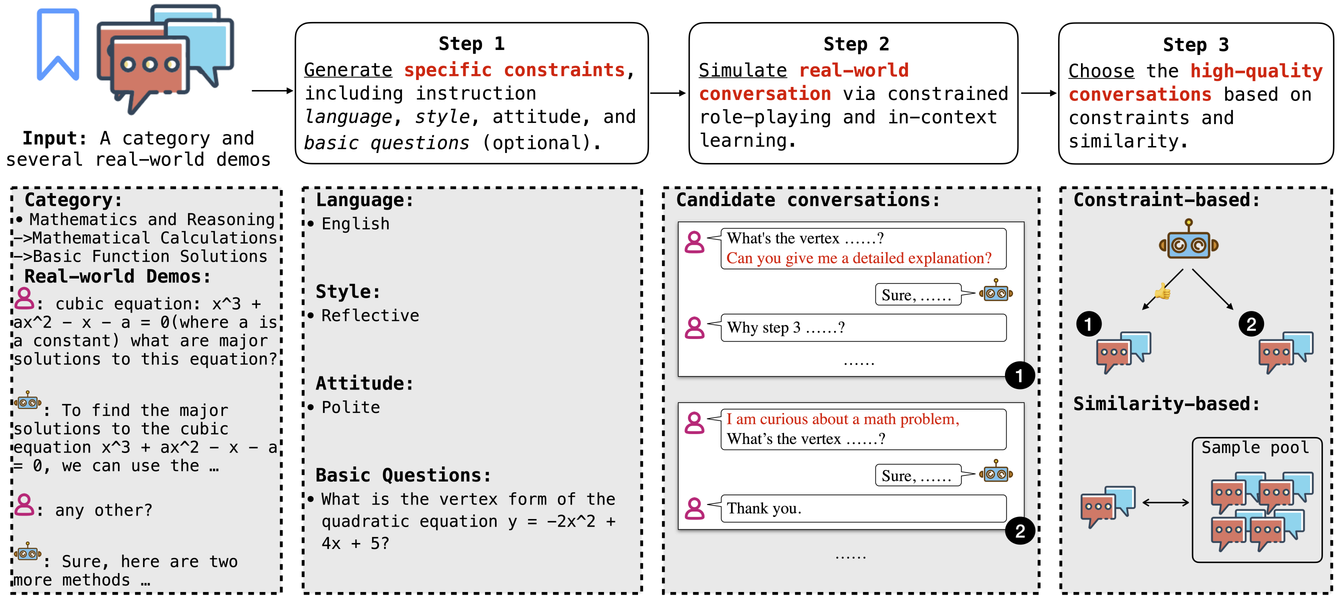

Input: A category and several real-world demos
Step 1
Generate specific constraints, including instruction language, style, attitude, and basic questions (optional).
Step 2
Simulate real-world conversation via constrained role-playing and in-context learning.
Step 3
Choose the high-quality conversations based on constraints and similarity.
Category:
• Mathematics and Reasoning
->Mathematical Calculations
->Basic Function Solutions
Real-world Demos:
: cubic equation: x^3 + ax^2 − x − a = 0(where a is a constant) what are major solutions to this equation?
: To find the major solutions to the cubic equation x^3 + ax^2 − x − a = 0, we can use the …
: any other?
: Sure, here are two more methods …
Language:
• English
Style:
• Reflective
Attitude:
• Polite
Basic Questions:
• What is the vertex form of the quadratic equation y = −2x^2 + 4x + 5?
Candidate conversations:
What's the vertex ……?
Can you give me a detailed explanation?
Sure, ……
Why step 3 ……?
……
1
I am curious about a math problem, What's the vertex ……?
Sure, ……
Thank you.
2
……
Constraint-based:
1
2
Similarity-based:
Sample pool

Figure 2: A high-level pipeline of Instruction Data Collection.

3.3 Instruction Data Collection

Generating diverse instructions is very challenging for human annotators, as it requires (1) the ability to transition between various instruction styles, attitudes, and languages, and (2) the capacity to produce a range of samples within a single category (*e.g.,* "Grammar-based Rewriting"). Consequently, we propose employing the highly capable LLM, GPT-4 (OpenAI 2023), to simulate a variety of user types and generate diverse, high-quality instructions for each category. Please note that we do not directly incorporate instruction data from ShareGPT into our benchmark, because numerous LLMs (*e.g.,* Vicuna (Chiang et al. 2023), TÜLU (Wang et al. 2023)) have already utilized data from shareGPT for supervised instruction-tuning. Therefore, we only use ShareGPT data as seed data to guide GPT-4. The data collection pipeline is depicted in Figure 2.

For any leaf category (*e.g.,* "Mathematics and Reasoning "→"Mathematical Calculations"→ "Algebraic Equation Problems"), we consider the following three steps to collect the instruction-following data.

In the first step, the goal is to generate constraints to guide GPT-4 to simulate specific user types, thereby preventing generation of unrealistic instructions. To achieve this, we treat the seed data as a sample pool and randomly select two samples as demos of in-context learning, each associated with a particular instruction style ($\mathcal{S}$), attitude ($\mathcal{A}$), and language ($\mathcal{L}$). We randomly sample target style, attitude, and language from these two demos to form constraints, compelling GPT-4 to learn from different instruction demos rather than excessively imitating one. For example, the constraints in Figure 2 is $\{\mathcal{S}=$"$Reflective$", $\mathcal{A}=$"*Polite*", $\mathcal{L}=$"*English*"$\}$. Additionally, given that GPT-4 may struggle to generate high-quality mathematical or logical reasoning questions, we gather data from previous task-specific benchmarks as basic questions, which are then incorporated as part of the constraints. For example, we use the GSM8K (Cobbe et al. 2021a) dataset as a basic question source for the "Mathematics and Reasoning" →"Word Problems" category. More details of the existing dataset resources included in DINGO are listed in Table 3.

Category	Basic Question Source
Word Problems	GSM8K (Cobbe et al. 2021a)
Mathematical Theorem	ProofNet (Azerbayev et al. 2023)
Combinatorics	Math (Hendrycks et al. 2021)
Numerical Calculation	Math (Hendrycks et al. 2021)
Common Sense Reasoning	StrategyQA (Geva et al. 2021)
Logical Reasoning	LogiQA (Liu et al. 2020)
Text2Code	MBPP (Austin et al. 2021)

Table 3: The basic question source of DINGO.

The goal of the second step is for GPT-4 to simulate a real-world user and generate high-quality instructions by adhering to the constraints. We use in-context learning to achieve this goal. As illustrated in Table 2, we combine the task description, the two demos obtained from the first step, and the target constraints as input context for GPT-4. As demonstrated, GPT-4 learns different expressions from two demos and transforms the basic question into specific instruction "*Act as a helpful math assistant to* . . . " based on the **Role-play** and **Command** constraints. Following previous work (Wiegreffe et al. 2021; Yuan et al. 2023), we adopt the *over-generate-then-filter* approach to obtain higher quality instructions. Thus, in this step, we prompt GPT-4 to make two predictions based on the same input, generating two instruction candidates.

In the third step, the objective is to select faithful and diverse instructions. We consider two selection methods, constraint-based pair-wise selection and similarity-based selection. Specifically, we first use GPT-4 to determine which of the two candidates adheres more closely to the constraints. We require GPT-4 to choose from three options, $\{first, second, tie\}$, and provide a rationale. Next, to ensure diversity, we calculate the similarity between the best candidate and the collected data in the dataset. Then, we only add the candidate to the dataset if the maximum ROUGE-L similarity is less than 0.6.

3.4 Dataset Analysis

Table 4 presents statistics of DINGO, which exhibits two main characteristics: (1) More fine-grained tasks are divided under each first-level category, such as “Biology” and “Chemistry” within the “Knowledge Utilization” category. (2) Each sample may comprise multiple turns of questions, simulating the process of human interaction with LLMs.

To validate diversity within each category, we calculate the overlap degree of instructions in each category. Figure 3 illustrates the similarity distribution of instructions. For each instruction, we compute its highest ROUGE-L score with regard to other instructions in the same category. The results illustrate the diversity of instructions in DINGO.

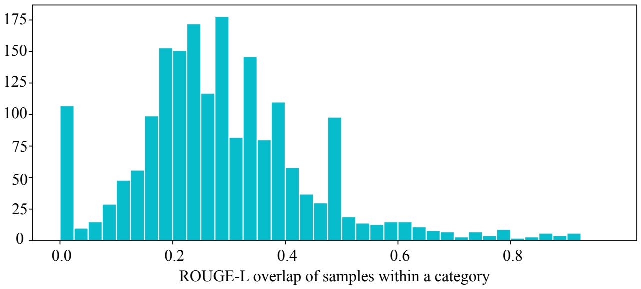


Figure 3: Distribution of the ROUGE-L scores between instructions within a category.

4 Experiments

4.1 Experimental Setup

Baseline Models We select two representative types of LLMs: (1) Pre-trained only LLMs, including Llama (Touvron et al. 2023a) and Llama2 (Touvron et al. 2023b); and (2) Instruction-tuned LLMs, including vicuna-v1.3 (Chiang et al. 2023), vicuna-v1.5 (Chiang et al. 2023), Llama2-chat (Touvron et al. 2023b). Considering that vicuna-v1.3 is instruction-tuned from Llama and vicuna-v1.5 is instruction-tuned from Llama2, we refer to vicuna-v1.3 as vicuna and vicuna-v1.5 as vicuna2 in this paper to make the notations consistent with Llama and Llama2.

Evaluation Method We employ the `LLM-as-a-judge` method to comprehensively evaluate LLM’s responses (Zheng et al. 2023). `LLM-as-a-judge` is a technique to score the performance of LLMs by utilizing GPT-4. Researchers have discovered that GPT-4 can generate consistent scores and provide detailed justifications, which exhibit a high level of agreement with human experts. However, considering that GPT-4 has difficulty in accurately scoring math/code problems (Cobbe et al. 2021b), we include the standard answers for basic questions as a reference in the prompt given to GPT-4. Regarding the grading method, `LLM-as-a-judge` considers two types, pair-wise comparison and single-answer grading. However, considering that we need to compare the performance of multiple LLMs, we choose to use single-answer grading for more efficient evaluation. For different categories, we have manually annotated different scoring criteria to assist GPT-4 in generating scores that align with human preferences. For instance, in “Mathematics and Reasoning” tasks, the primary considerations include the clarity of steps, the correctness of reasoning, and the appropriateness of natural language explanations. Meanwhile, for “Knowledge Unilization” tasks, the primary considerations is on the adequacy of key points and whether the answers contain hallucination.

We explore the agreement between these two grading methods and human experts in Section 4.2.

4.2 Experimental Results

How do the existing LLMs perform on DINGO? Figure 4-(a) shows the overall performance of ten LLMs on the first-level categories of DINGO. First, comparing pre-trained LLMs with instruction-tuned LLMs, such as `Llama-13B` and `vicuna-13B`, we can see that instruction-tuning significantly impacts alignment with human preferences. Second, comparing different instruction-tuned LLMs based on the same pre-trained LLMs, such as `vicuna2-7B` and `Llama2-chat-7B`, we find that `Llama2-chat-7B` has better instruction-following ability than `Vicuna2-7B`. This is mainly because `Llama2-chat-7B` utilizes an RLHF (reinforcement learning from human feedback) framework with two reward models for usefulness and safety to align with human preferences, enabling it to outperform the base LLM (*i.e.,*`Llama2-7B`) under various user instructions. Finally, comparing LLMs of different sizes indicates that increasing the model size significantly improves the instruction-following ability of the pre-trained LLMs (such as `Llama2-7B` and `Llama2-13B`), but the impact on instruction-tuned LLMs (such as `Llama2-chat-7B` and `Llama2-chat-13B`) is comparatively weaker.

Can instruction-tuning consistently achieve stable improvements in more fine-grained categories? Figure 4-(b) illustrates the performance across all sub-categories under “Knowledge Utilization”→“Open-Book Questions”→“Knowledge-Intensive Questions”. It can be seen that under a more fine-grained evaluation, the improvement brought by instruction-tuning is not consistent. For example, the instruction-following performance of `vicuna2-7B` after instruction-tuning does not improve compared to its base LLM `Llama2-7B` in the two sub-categories: “Biology” and “Medicine”. This suggests that conducting a more fine-grained evaluation of LLMs’ instruction-following ability is necessary, as high scores in coarse categories (*e.g.,* “Knowledge Utilization”) do not

Category	#Tasks	#Samples	#Turns	#Input length
Mathematics and Reasoning	9	432	1.6	83.4
Language Understanding	11	530	2.0	125.7
Language Generation	14	730	1.9	157.1
Knowledge Utilization	34	1596	2.6	72.7
Creation	31	1498	1.9	63.4
Code	5	240	2.1	105.5

Table 4: The statistics of the DINGO dataset. ‘Category’ represents the first-level category in the category tree. ‘#Tasks’ represents the number of tasks belonging to each first-level category. ‘#Samples’ represents the number of samples contained in each first-level category. ‘#Turns’ represents the average number of conversation turns included in each sample. ‘#Input length’ represents the average length of user input in each sample.

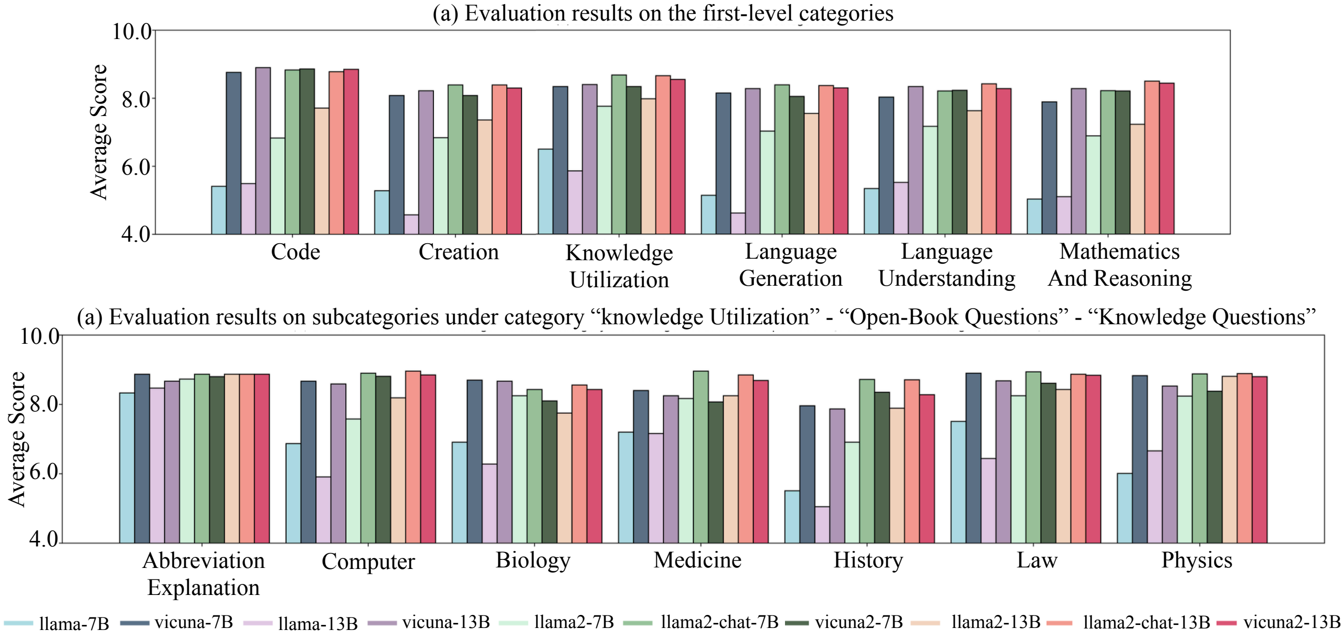


Figure 4: Evaluation results of different LLMs under different category granularity.

necessarily indicate stable performance in all finer subcategories (*e.g.,* “Biology”). Therefore, DINGO helps guide instruction-tuned LLMs towards a more comprehensive direction for improvement, thus enhancing the interactive experiences for users with diverse requests.

Do diverse instruction types pose a challenge to LLMs? To investigate whether the diverse instruction types in DINGO present a significant challenge to LLMs, we conduct an analysis on the categories containing basic questions. Specifically, we use four LLMs to respond to basic questions and instructions in DINGO across four subcategories. The experimental results are shown in Figure 5. It can be observed that the instruction following scores of the four LLMs on DINGO are lower than those on basic questions, indicating that the diverse instructions in DINGO are more challenging compared to standard questions. This also suggests that it is necessary to evaluate the LLMs’ instruction following ability using more diverse instructions, as an LLM may perform well in one mode of expression but not in others, implying that the LLM’s robustness to diverse instructions in real-world scenarios might be insufficient.

Additionally, to intuitively understand why the LLMs perform poorly on diverse instructions, we present two examples in Table 5. Example-1 indicates that when user instructions become more concise and require a concise output (*i.e.,* “*Directly output the answer without any explanation.*”), LLMs still generate lengthy explanations that do not align with user instructions. Example-2 shows that when the instruction is in **Challenge** style (*i.e.,* “*Please check each condition carefully!*”), the LLMs may go against the original correct answer in order to cater to human users, *i.e.,* “*Therefore, the final answer is None of the above.*”.

What is the agreement between human judge scores and GPT-4 judge scores? To evaluate the agreement between GPT-4 and human experts, we choose 100 examples from DINGO and employ six human experts. Given a judge (*i.e.,* either GPT-4 or human expert), we ask the judge to score the responses of the LLMs using two methods, (1) pairwise comparison and (2) single-answer grading. Pairwise comparison provides the judge a question and two potential an-

From	DINGO	Basic Question
Example-1	User: Find out the number of ways Kim can place 10 identical lamps on 3 identical tables. Directly output the answer without any explanation. LLM: There are several ways to calculate the number of ways Kim can place 10 identical lamps on 3 identical tables . . .	User: Kim has 10 identical lamps and 3 identical tables. How many ways are therefor her to put all the lamps on the tables? LLM: First, let's count the ways to put 2 identical lamps on a single table. There are 2! ways to arrange . . .
Example-2	User: For listed companies, . . . If the above statement is true, which statement cannot be true? A.A listed company needs financing and has no way to pay dividends. B . . . C . . . D . . . LLM: Let's analyze each statement one by one: . . . Therefore, the final answer is D. User: Please check each condition carefully! LLM: I apologize for any confusion . . . Therefore, the final answer is None of the above.	User: For listed companies, . . . If the above statement is true, which statement cannot be true? A. A listed company needs financing and has no way to pay dividends. B . . . C . . . D . . . LLM: Let's analyze each statement one by one: . . . Therefore, the final answer is D.

Table 5: Two illustrative examples demonstrate why the DINGO is more challenging than basic questions in instruction following. We use ellipses to omit lengthy, less significant sentences.

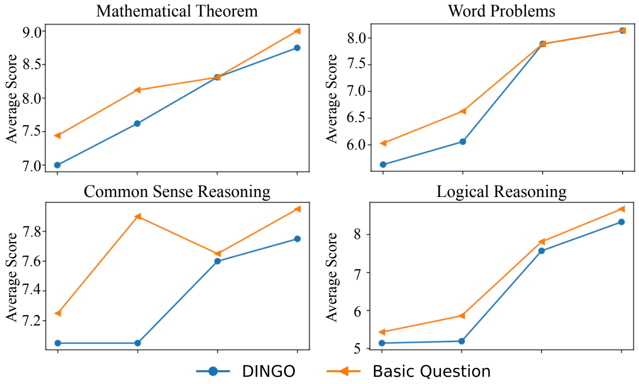
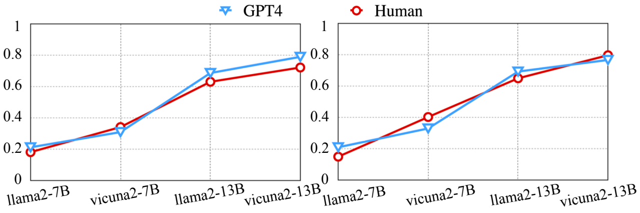


Figure 5: Comparison of instruction following performance of LLMs on DINGO and on basic questions.

swers, and asks the judge to decide which answer is more appropriate. Single-answer grading asks a juedge to assign a score to a specific answer. Figure 6 shows the agreement between GPT-4 and humans under the two scoring methods. With pairwise comparison, GPT-4 has higher agreement with human. However, pairwise comparison would incur high cost. On the other hand, single-answer grading is more efficient. Thus, we recommend single-answer grading for rough identification of model issues, and pairwise comparison for more detailed evaluations.

5 Related Work: Evaluation of LLMs

For benchmarking the effectiveness of LLMs, various evaluation frameworks have emerged. Frameworks such as HELM (Liang et al. 2022) and BIG-BENCH (bench authors 2023) focus on the effectiveness of LLMs on a wide range of NLP tasks, mainly evaluating the problem solving ability of the model, without paying attention to the LLM's instruction-following ability. Recently, some work has started to focus on the instruction-following ability of LLMs. For example, InstructEval (Chia et al. 2023) focuses on evaluating the ability of Instruction-Tuned LLMs on three aspects, including problem solving, writing, and alignment. Alpaca Farm (Dubois et al. 2023) and Chatbot Arena (Zheng et al. 2023) focus on evaluating the open-ended instruction-following ability of LLMs. However, there are two main differences between DINGO and the above studies: (1) a diverse set of instructions based on real-world scenarios, which can comprehensively evaluate the model's instruction-following performance. (2) a fine-grained task category tree, which can deeply analyze LLM's instruction-following ability on fine-grained task types and pinpoint the deficiencies for further improvement.



(a) Pairwise Comparison

(b) Single-Answer Grading

Figure 6: Average win rate of four LLMs under different judge methods.

6 Conclusion

In this paper, we have presented a diverse and fine-grained instruction-following evaluation dataset DINGO. Based on a multi-level category tree with 130 nodes derived from real-world user requests, DINGO includes 5026 diverse instructions. Our experiments demonstrate that (1) while an instruction-tuned LLM may excel in broad categories, its performance can vary in fine-grained categories; (2) diverse instructions pose greater challenges for LLMs to generate responses that match human preferences.

Acknowledgments

This paper is supported by the Science and Technology Development Fund of Macau SAR (File no. 0081/2022/A2, 0123/2022/AFJ, and 0015/2019/AKP), and GuangDong Basic and Applied Basic Research Foundation (No. 2020B1515130004). This paper is also partly supported by the NSF of China (62122090, 62072461 and 62072458), the Fund for Building World-Class Universities (Disciplines) of Renmin University of China, the Beijing Natural Science Foundation (L222006), and the Research Funds of Renmin University of China.

References

Austin, J.; Odena, A.; Nye, M. I.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C. J.; Terry, M.; Le, Q. V.; and Sutton, C. 2021. Program Synthesis with Large Language Models. *CoRR*, abs/2108.07732.

Azerbayev, Z.; Piotrowski, B.; Schoelkopf, H.; Ayers, E. W.; Radev, D.; and Avigad, J. 2023. ProofNet: Autoformalizing and Formally Proving Undergraduate-Level Mathematics. *CoRR*, abs/2302.12433.

bench authors, B. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S. M.; Nori, H.; Palangi, H.; Ribeiro, M. T.; and Zhang, Y. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Chia, Y. K.; Hong, P.; Bing, L.; and Poria, S. 2023. INSTRUCTEVAL: Towards Holistic Evaluation of Instruction-Tuned Large Language Models. *arXiv preprint arXiv:2306.04757*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021a. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. *ArXiv*, abs/1903.00161.

Dubois, Y.; Li, X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Trans. Assoc. Comput. Linguistics*, 9: 346–361.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In Vanschoren, J.; and Yeung, S., eds., *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Liu, J.; Cui, L.; Liu, H.; Huang, D.; Wang, Y.; and Zhang, Y. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 3622–3628. ijcai.org.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Wang, Y.; Ivison, H.; Dasigi, P.; Hessel, J.; Khot, T.; Chandu, K. R.; Wadden, D.; MacMillan, K.; Smith, N. A.; Beltagy, I.; and Hajishirzi, H. 2023. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. *CoRR*, abs/2306.04751.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022. Emergent Abilities of Large Language Models. *Trans. Mach. Learn. Res.*, 2022.

Wiegreffe, S.; Hessel, J.; Swayamdipta, S.; Riedl, M.; and Choi, Y. 2021. Reframing human-AI collaboration for generating free-text explanations. *arXiv preprint arXiv:2112.08674*.

Ye, S.; Kim, D.; Kim, S.; Hwang, H.; Kim, S.; Jo, Y.; Thorne, J.; Kim, J.; and Seo, M. 2023. FLASK: Fine-grained Language Model Evaluation based on Alignment Skill Sets. *CoRR*, abs/2307.10928.

Yuan, S.; Chen, J.; Fu, Z.; Ge, X.; Shah, S.; Jankowski, C. R.; Yang, D.; and Xiao, Y. 2023. Distilling Script Knowledge from Large Language Models for Constrained Language Planning. *arXiv preprint arXiv:2305.05252*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; Du, Y.; Yang, C.; Chen, Y.; Chen, Z.; Jiang, J.; Ren, R.; Li, Y.; Tang, X.; Liu, Z.; Liu, P.; Nie, J.; and Wen, J. 2023. A Survey of Large Language Models. *CoRR*, abs/2303.18223.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*.