#INSTAG: INSTRUCTION TAGGING FOR ANALYZING SUPERVISED FINE-TUNING OF LARGE LANGUAGE MODELS

Keming Lu* **& Hongyi Yuan*†& Zheng Yuan & Runji Lin†**
Alibaba DAMO Academy
{lukeming.lkm,yuanhongyi.yhy,yuanzheng.yuanzhen,linrunji.lrj}@alibaba-inc.com

Junyang Lin & Chuanqi Tan & Chang Zhou & Jingren Zhou
Alibaba DAMO Academy
{junyang.ljy,chuanqi.tcq,ericzhou.zc,jingren.zhou}@alibaba-inc.com

ABSTRACT

Foundation language models obtain the instruction-following ability through supervised fine-tuning (SFT). Diversity and complexity are considered critical factors of a successful SFT dataset, while their definitions remain obscure and lack quantitative analyses. In this work, we propose INSTAG, an open-set fine-grained tagger, to tag samples within SFT datasets based on semantics and intentions and define instruction diversity and complexity regarding tags. We obtain 6.6K tags to describe comprehensive user queries. We analyze popular open-sourced SFT datasets and find that the model ability grows with more diverse and complex data. Based on this observation, we propose a data selector based on INSTAG to select 6K diverse and complex samples from open-source datasets and fine-tune models on INSTAG-selected data. The resulting models, TAGLM, outperform open-source models based on considerably larger SFT data evaluated by MT-BENCH, echoing the importance of query diversity and complexity. We open-source INSTAG in https://github.com/OFA-Sys/InsTag.

1 INTRODUCTION

The rise of contemporary chatbots, including GPT-4 (OpenAI, 2023), has brought to the forefront of generative artificial intelligence that is based on large language models (LLMs) to tackle a variety of real-world tasks. Well-aligned LLMs with human expectations can precisely recognize human intentions and properly formalize responses expressed in natural languages (Wang et al., 2023d). Achieving such a level of alignment typically necessitates fine-tuning processes, such as supervised fine-tuning (SFT) (Taori et al., 2023; Chiang et al., 2023; Touvron et al., 2023b), response ranking (Yuan et al., 2023b; Song et al., 2023; Rafailov et al., 2023), and reinforcement learning with human feedback (RLHF) (Bai et al., 2022a; Ouyang et al., 2022; Touvron et al., 2023b), to enable LLMs to comprehend and execute diverse instructions effectively.

A broad range of training data covering various semantics and specialties is crucial for achieving alignment with human preference through SFT, which is typically gathered through crowd-sourcing (Ouyang et al., 2022; Bai et al., 2022a; Touvron et al., 2023b) or by distilling from other LLMs (Taori et al., 2023; Ding et al., 2023). The SFT data for alignment is generally formalized in a multi-turn utterance manner, and each turn is composed of a human query and a corresponding response expected to generate by well-aligned chatbots. Recent research indicates that the training dataset for alignment should be diverse and complex, covering various domains, tasks, and formats (Xu et al., 2023a; Mukherjee et al., 2023; Wang et al., 2023b). Such diversity and complexity are mainly determined by query formation. Various methods are proposed and claimed to improve the diversity and complexity of the queries and advance the alignment of LLMs (Wang et al. 2023c; Xu

*Equally Contributed. Order determined by swapping the one in Yuan et al. (2023a).
†Work done during internships at Alibaba DAMO Academy.

et al. 2023a; Ding et al. 2023; *inter alia*). However, how to quantify the diversity and complexity of queries is significantly understudied and hence less analyzed.

To shed light on this topic, we propose using a tagging system to feature and categorize samples in SFT datasets. Given the versatile tasks aligned LLMs are expected to handle, an equally versatile tag system is necessary to distinguish open-world human queries. However, building an open, fine-grained tagging system manually is infeasible to scale for large datasets. To this end, we propose INSTAG, an automatic INStruction TAGging method empowered by proprietary high-performing chatbot CHATGPT[1]. Leveraging such a well-aligned chatbot, INSTAG designs a framework to prompt CHATGPT to automatically assigns tags to queries. INSTAG achieves the increased quality of the tag assignment by deliberately prompting CHATGPT to explain each tag assigned and including a systematic tag normalization procedure. We apply INSTAG to a collection of existing rich open-source SFT datasets and build open-set, fine-grained tags which, as we observed, can reflect the semantics and intentions beneath human queries in SFT datasets. Through the scope of tags, we conduct a detailed and quantified analysis of existing open-source datasets, providing insights into query distributions in terms of diversity and complexity. Such analyses reveal that diverse and complex queries induce high alignment performance during SFT. Following this insight, we propose a data selector based on INSTAG, including a complexity-focus diverse sampling method that can extract the most complex queries in a diverse distribution. LLMs fine-tuned with data selected by the INSTAG selector perform well on the popular benchmark MT-BENCH(Zheng et al., 2023), supporting our previous query distribution insights.

The contributions of this work are mainly three-fold. Firstly, we propose using open-set fine-grained intention tags as instruction diversity and complexity metrics. To this end, we develop INSTAG, an annotator that leverages the instruction-following abilities of proprietary chatbots and employs tag normalization methods. Secondly, we analyze existing open-source SFT datasets and provide insights into query diversity and complexity. Finally, we design a data selector based on INSTAG and apply it to the latest open-source datasets. The resulting best LLMs, TAGLM-13b-v1.0 and TAGLM-13b-v2.0 respectively based on LLaMA (Touvron et al., 2023a) and LLaMA-2 (Touvron et al., 2023b), aligned with selected data achieve scores of 6.44 and 6.55 on the benchmark MT-BENCH, respectively, surpassing a group of LLMs aligned with considerably more SFT data. Our contributions are verified with experiments and multifaceted analysis. Most notably, INSTAG exhibits its robust potential to offer deeper insights into aligning LLMs, extending beyond the data selection demonstrated in our work.

2 RELATED WORKS

LLMs for Human Alignment. Through supervised fine-tuning (SFT), response ranking, or reinforcement learning (Ouyang et al., 2022; Bai et al., 2022a;b; Yuan et al., 2023b; Rafailov et al., 2023; Song et al., 2023; Touvron et al., 2023b), LLMs can obtain versatile abilities for understanding and following diversified human queries expressed in natural languages, aligning with human intentions. Recent research mainly focused on SFT to align LLMs with human intentions and has contributed essential practices to developing open-resourced well-aligned LLMs, which is adequately summarized by Zhao et al. (2023). Several prominent works collected SFT data through human annotated demonstrations (Ouyang et al., 2022; Touvron et al., 2023b), online user logs of proprietary LLMs (Chiang et al., 2023; Wang et al., 2023a; Köpf et al., 2023), or prompting proprietary high-performing LLMs such as CHATGPT or GPT-4 (OpenAI, 2023) to generate or rewrite samples (Taori et al. 2023; Ding et al. 2023; Xu et al. 2023a; Mukherjee et al. 2023; *inter alia*). Different LLMs fine-tuned on the datasets have aligned with human preference and exhibited good performance in various real-world tasks.

Data for Human Alignment. It has been highlighted that the performance of aligned LLMs is affected by the quality of the SFT data. Such data quality pertains to the level of responses (Peng et al., 2023; Chiang et al., 2023), the difficulty of tasks presented (Mukherjee et al., 2023), the complexity of queries (Xu et al., 2023a), the diversity of semantics (Ding et al., 2023; Taori et al., 2023), and the scale of sample amounts (Zhou et al., 2023). Taori et al. (2023) used Self-Instruct (Wang et al., 2023c) to generate diversified queries for SFT and Xu et al. (2023a) proposed Evol-Instruct

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

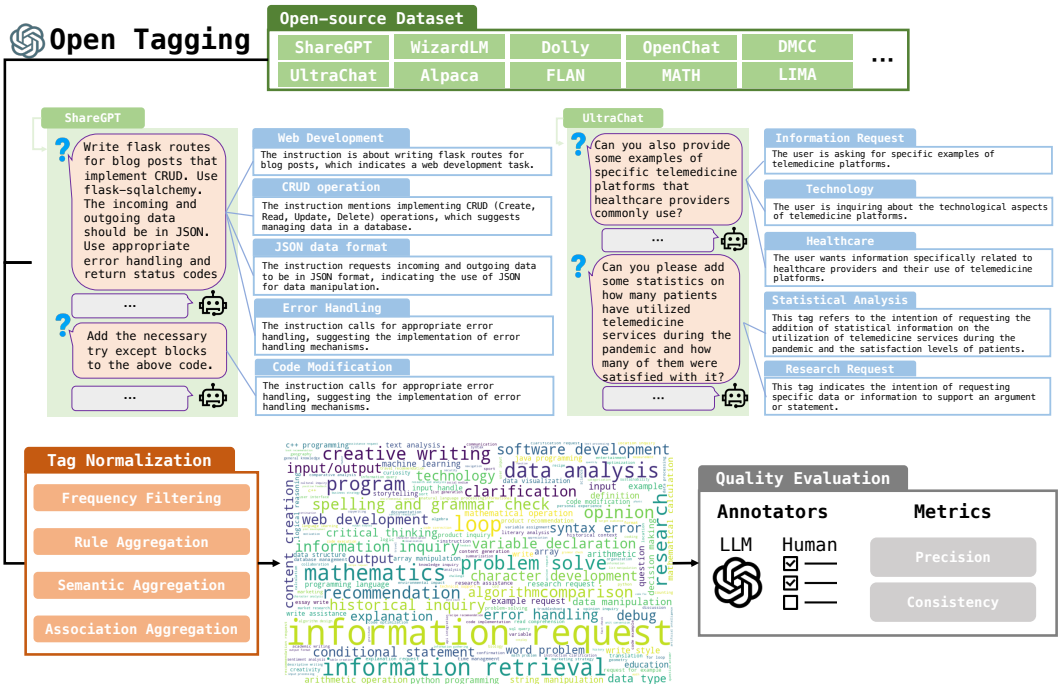


Figure 1: Overview of INSTAG. We use CHATGPT to annotate fine-grained intention tags for a series of open-source datasets. We present two cases of open tagging annotations in this figure, selected from ShareGPT and UltraChat. A tag normalization, including multiple denoising and aggregation methods, is then applied to the original tagging results. Finally, the quality of the tag set, as shown in the word cloud, is evaluated by human and LLM annotators, focusing on the tagging precision and consistency.

to complexify simple queries for better human alignment. Mukherjee et al. (2023) used proprietary high-performing LLMs to rewrite the queries and responses of samples from FLAN collection (Longpre et al., 2023) and observed improvement of LLMs in conventional NLP task solving. Ding et al. (2023) proposed UltraChat using manually designed diverse anchor concepts and entities to generate multi-turn data by inducing conversations in CHATGPT. OpenChat (Wang et al., 2023a) and Vicuna (Chiang et al., 2023) are both current open-sourced LLMs with cutting-edge instruction following abilities, and both models are trained on the user logs of GPT-4 from ShareGPT [2]. As evaluated in Wang et al. (2023b), the success of fine-tuning on ShareGPT demonstrates that queries from user logs are of higher diversity and the responses generated from GPT-4 are of better quality, resulting in superior instruction following the abilities. Zhou et al. (2023) found that a small amount of high-quality data is sufficient for LLMs to excel at human alignment.

Although current research proposed more diversified and complexified SFT data and made significant progress in developing well-aligned LLMs with human intentions, existing works have yet to discuss how to quantify the diversity and complexity of queries. Taking advantage of the high-performing CHATGPT and GPT-4, we annotate existing data samples with tag entities. Through the scope of tags, we quantify the diversity and complexity of the training data for the first time and study the data mixture for better alignment.

3 INSTAG

This section presents an automatic instruction tagging method to identify proper tags describing instruction intentions in an open setting. We first define fine-grained intention tags and present the open tagging process with LLMs (§3.1). Then, we design a systematic normalization method to denoise the raw tags from previous annotations (§3.2). We also fully evaluate the tagging quality to

[2] https://sharegpt.com/

Inconsistency	Examples	Output
Lexical Noise **Rule Aggregation**	Information Retrieval, information_retrieval, infomation retrieve	information retrieval
Uncontrolled Granularity **Semantic Aggregation**	information request, request for information, request for additional information, request for more information, additional information request, specific information request	information request
Spurious Correlations **Association Aggregation**	(mathematics, math problem), (loop, for loop)	mathematics, for loop

Table 1: Inconsistency in intention tagging results from open-set annotations. Inconsistencies can be addressed with three aggregation methods described in §3.2.

ensure INSTAG generates precise and consistent intention tags (§3.3). Finally, we use INSTAG to analyze open-source SFT datsets (§3.4).

3.1 OPEN-SET FINE-GRAINED TAGGING

Instructions, or queries in the context of modern chatbots, serve as expressions of user intentions, which can often be multifaceted and highly intricate. To illustrate, we showcase an instruction from the ShareGPT dataset in Fig. 1, where the user submits a coding request specifying desired output formats and error-handling methods. To better parse such instructions, it is essential to employ fine-grained tags that can identify atomic intentions rather than relying on generalized, coarse-grained classes. However, although fine-grained intention tags offer a more detailed understanding of instruction distribution, they also present challenges in annotation and normalization. Therefore, we propose an open-set tagging with CHATGPT and a normalization technique to address these issues. In other words, we do not provide a predefined ontology of tags when annotating with CHATGPT. We choose an open setting since a closed set is not flexible enough to cover versatile intentions in open chatting.

We use the prompt shown in Tab. 4 to employ CHATGPT, providing fine-grained intention tags for given queries. We provide few-shot examples in the prompt to hint CHATGPT provides tags in a specific format and requires CHATGPT to provide JSON outputs for parsing. As shown in Fig. 1, we separately annotate each query in a chat session and require CHATGPT to briefly explain tags for the convenience of quality evaluation. The number of original tags annotated by CHATGPT is larger than 12 thousand, showing CHATGPT can provide diverse and fine-grained annotations. However, we notice the original tagging results provided by CHATGPT contain noticeable noises, including inconsistency in word format and granularity. Therefore, we design a systematic method to normalize the open-set tagging results from CHATGPT.

3.2 TAG NORMALIZATION

The production of intention tags through CHATGPT in an open setting presents a challenge in ensuring consistency, as no predefined ontology is provided, resulting in noise in the raw tagging outcomes. We have identified three significant types of noise, detailed in Tab. 1, which have the potential to impact downstream processes: **Lexical Noise**, arises from the instability of CHATGPT in adhering to output format instructions and can be mitigated through the use of stemming as a post-processing step; **Uncontrolled Granularity** refers to the potential for CHATGPT to produce tags that are overly specific, such as "request for more information" as shown in Tab. 1; **Spurious Correlations** refer to tags that often appear together due to the bias of CHATGPT or data distributions. Such tag groups should be merged to form an atomic tag. These issues must be addressed to ensure that intentions are accurately identified and utilized in downstream processes. Therefore, we normalize open-set tagging results by various aspects, including frequency, format, semantics, and associations. Specifically, we clean the raw tagging results with the following normalization procedure:

Metric	GPT-4 Annotation	Human Annotation (1%)	Agreement (κ)	
			Human-Human	Human-GPT
Tag Precision	96.1	100	0.47	0.92
Tag Consistency	86.6	100	0.73	0.75

Table 2: Evaluation for the tagging quality of INSTAG. We design two metrics, tagging precision and consistency, for evaluating INSTAG. We employ GPT-4 to label 4,000 tagging results. Moreover, we also employ three human annotators to annotate 1% cases and report their majority voting. We report agreement between human annotators in Fleiss-kappa scores and agreement between majority voting and GPT-4 in Cohen's kappa scores. We also create counterfactual cases to probe the judgment abilities of different annotators shown in Tab. 8.

- **Frequency Filtering**: We first filter out long-tail tags appearing less than α times in the whole annotated dataset. α is a hyperparameter related to the scale of the dataset.
- **Rule Aggregation**: We transform all tags into lower characters to avoid the influence of capitalization. We also replace all special characters into spaces to further aggregate the tags. Finally, we apply stemming to each tag with the support of NLTK (Bird et al., 2009).
- **Semantic Aggregation**: We employ text embedding models to obtain the semantics of tags. We use PHRASEBERT (Wang et al., 2021), a BERT-based model designed explicitly for embedding phrases, such as titles of tags. Other embedding methods, such as OpenAI embeddings or DENSEPHRASE (Lee et al., 2020), can also be adopted as alternatives. After we obtain the semantic embeddings of tags, we use DBSCAN algorithm (Hahsler et al., 2019) to cluster tags with a given threshold t of semantic similarity. Similarly, other density clustering methods can be used instead of DBSCAN for the same denoising purpose. Semantic aggregation controls the granularity of tags in terms of semantic similarities.
- **Association Aggregation**: We notice CHATGPT tends to provide highly associated tags that are expected to consider as an atomic tag as a whole, which mainly occurs in mathematics and coding queries. Therefore, we analyze all raw tagging results and employ the FP-Growth algorithm (Han et al., 2000) to mine association rules between tags. We then recursively merge associated tags based on the above association rules and reduce verbosity.

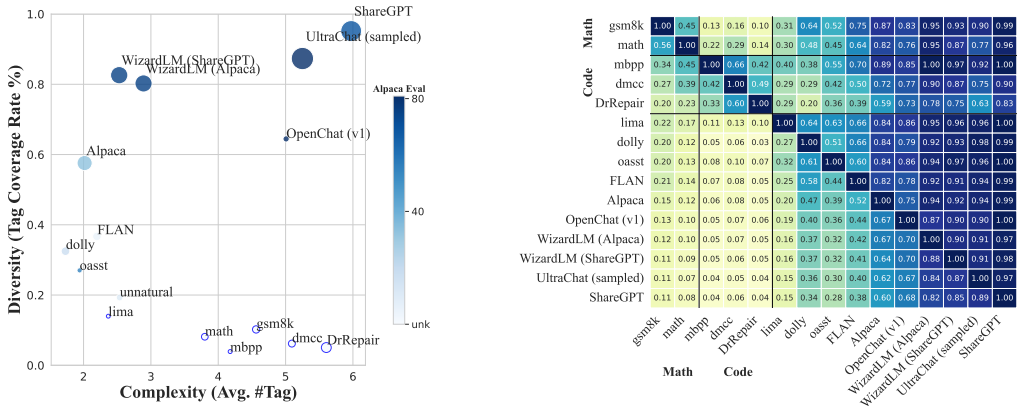
We apply INSTAG on widely-used open-source SFT datasets introduced in Appx. §C with details. Over 100 thousand original unique tags are generated following the CHATGPT annotation. To filter out long-tail cases, we implement Frequency Filtering with a value of $\alpha = 20$, resulting in the retention of 8,541 entities. We apply the rule aggregation to address lexical noise, which merged tags and reduced the count to 7,157. We then utilize semantic aggregation, implementing DBSCAN clustering with a minimum semantic similarity of 0.05, to further merge and reduce the count to 6,587 tags. Finally, we employed the association aggregation with a minimum support of 40 times and a minimum confidence of 99%, producing 1,772 association rules to transform tag groups into atomic tags. These measures were essential in streamlining the tagging process and ensuring the quality of downstream processes.

An overview of these six thousand tags locates in Appx. §A. We also train a local specialized tagging LLM, INSTAGGER, based on normalized tagging data to distill such annotation abilities into smaller LLMs, shown in Appx. §I.

3.3 QUALITY EVALUATION

We employ both GPT-4 and human annotators to provide judgments in a set of randomly sampled tagging results. We evaluate the quality of the normalized tagging dataset in precision and consistency:

Precision. We define precision as whether tags assigned to a specific query are all correctly related to query intentions. Tag precision is essential since fine-grained tags should be precisely expressed as part of query intentions. For example, given a case $(q, \mathcal{T})$ where q is the query and $\mathcal{T}$ is tags assigned to it, we employ annotators to identify any incorrect tags in $\mathcal{T}$. We consider it a negative case if any tag in $\mathcal{T}$ is annotated as incorrect. Otherwise, it is a precise tagging case.



(a) Diversity and Complexity based on Tags

(b) Dataset Correlation (Column recalls Row)

Figure 2: Open-source SFT dataset analysis based on tags. Fig. 2a shows diversities and complexities of various open-source SFT datasets based on metrics from tags, where datasets are marked in scatter sizes, and model performance on AlpacaEval (Li et al., 2023) are marked in scatter colors. The AlpacaEval performance is collected from the official leaderboard and Wang et al. (2023b), datasets without AlpacaEval scores are marked in circles. Fig. 2b displays correlations among SFT datasets based on the recalls of tags. Annotations are the ratio using unique tags of the dataset in the column that recalls the dataset in the row.

Consistency. To form a consistent tag ontology, we naturally require that the semantics of a specific tag will not shift across queries. An annotation case in consistency $(t, \mathcal{I})$ contains a tag t and a set of randomly selected instructions $\mathcal{I}$ annotated with such tag. Annotators are required to identify any semantic changes in tags across all instructions.

To be more specific, we randomly sample 4,000 cases, 2,000 each for precision and consistency, for GPT-4 annotation. Then, we hire three annotators to manually label 40 cases (1%) selected from the above evaluation set. Annotation from human annotators provides judgments and reveals confidence of GPT-4 annotation on a larger scale.

The evaluation results are shown in Tab. 2. GPT-4 provides 96.1% and 86.6% accuracy in tag precision and consistency, respectively. Meanwhile, we also report the majority voting of human annotators, which suggests a hundred-percent correctness among both tasks. We notice the Fleiss-kappa between human annotators reaches the basic agreement. In contrast, Cohen's kappa between majority voting and GPT-4 reaches more than 0.7, suggesting a solid agreement between human and GPT-4 annotators. To eliminate the possibility that such results contain robust false positive annotations, we specifically design counterfactual annotation experiments shown in Tab. 8 and proof that both human and GPT-4 are capable of precisely recalling incorrect cases. Therefore, tags provided by INSTAG are of good quality regarding precision and consistency for downstream analyses.

3.4 PRELIMINARY ANALYSIS

We present the preliminary analysis of existing open-source datasets through normalized tags in Fig. 2. To start with, we introduce the diversity and complexity attributes of SFT datasets induced by our tagging result:

- **Diversity** is used to access the range of intentions and semantics covered by queries in a dataset. According to the tagging results, a dataset is considered more diverse if it covers more individual tags. The attribute is quantified as the unique tag coverage rate for the overall tag set.
- **Complexity** aims to measure the number of intentions and semantics complicating queries. We assume a more complex query would be assigned more tags. The attribute is quantified as the average tag number assigned to queries in a dataset.

We first depict the overall assessments of each dataset regarding the axis of diversity and complexity as shown in Fig. 2a. Each dataset is represented as a dot whose size indicates the dataset sample size, and color indicates the performance of LLMs fine-tuned thereon. As shown, (1) **The larger size, the more diverse.** On the diversity axis, the larger datasets contain human queries of higher diversity. The dataset for OpenChat-v1 is filtered from ShareGPT, resulting in high query diversity and complexity while having a relatively small size. (2) **The larger size, the more complex.** On the complexity axis, we can see that WizardLM (Alpaca) has a larger average tag number than the Alpaca dataset; hence is more complex. WizardLM (Alpaca) is created by complicating the queries from Alpaca datasets using Evol-Instrcut. This observation demonstrates that the average tag number can present the diversity of an SFT dataset. The complexity is also positively correlated with data size, except for mathematical reasoning and code generation. (3) **Math and Code show different trends.** The datasets for mathematical reasoning (MATH, GSM8K) and code generation (DMCC, MBPP, DrRepair) focus on specific downstream abilities and result in low diversity, while such datasets have relatively high complexity. (4) **More diverse and complex data induces higher performance.** ShareGPT, UltraChat, and OpenChat-v1 datasets lay at the upper-right corner of Fig. 2a, having both high diversity and complexity. Vicuna, UltraChat, and Openchat, which are respectively fine-tuned on the datasets, achieve cutting-edge performance among open-resourced models in aligning with human preference, as evaluated by public leaderboards (e.g., AlpacaEval (Li et al., 2023)). This scenario verifies that LLMs can benefit from fine-tuning more diverse and complex data for alignment.

We demonstrate the correlations between datasets regarding unique tag recalls to understand the correlations between existing SFT datasets. As depicted in Fig. 2b, we use the tag sets of the datasets on each column to calculate the recall with respect to the tag sets of the datasets on each row. We can conclude from the figure that (1) **Tags can identify different tasks.** Datasets for mathematical reasoning and code generation tasks exhibit higher tag recalls within the tasks. This demonstrates that the tags can identify the uniqueness of mathematical reasoning and code generation datasets compared to more general-purpose datasets. (2) **One covers all.** WizardLM (Alpaca), WizardLM (ShareGPT), UltraChat, and ShareGPT, respectively, have higher tag recalls for other datasets. This demonstrates that the four datasets contain more diversified queries and cover other datasets, consistent with the results in Fig. 2a.

Overall, INSTAG provides a good tool for analyzing SFT datasets through the perspective of tagging. Existing SFT datasets differ in diversity and complexity as evaluated by the tagging results.

4 INSTAG FOR DATA SELECTION

As analyses shown in §3.4, we notice fine-tuning LLMs on more diverse and complex datasets may benefit alignment performance. Therefore, we present a data selection method supported by INSTAG in this section and align LLMs with selected data to show the effectiveness of INSTAG. We introduced experimental setup (§4.1), results (§4.2), and primary analyses related to query diversity and complexity (§4.3).

4.1 EXPERIMENT SETUP

Based on the normalized tagging results and the preliminary analyses of existing datasets as presented in Figure 2, we conduct fine-grained experiments to discuss the impact of data complexity and diversity through control variate methods. Under the correlation analyses in Figure 2b, each dataset of WizardLM(Alpaca), WizardLM(ShareGPT), UltraChat, and ShareGPT maintains large tag recalls regarding other datasets. The four datasets also have the largest average tag numbers shown in Figure 2a. These results indicate that the four datasets have high data diversity and complexity. Therefore, we pool the four datasets and create different subsets for delve-deep data complexity and diversity analysis. The pooled dataset contains 306,044 samples with a tag set size of 6,398 and an average tag number of 4.48.

In the following experiments, all the models fine-tuned are based on 13B version LLMs of either LLaMA (Touvron et al., 2023a) or LLaMA-2 (Touvron et al., 2023b). If not specified otherwise, we fine-tune the model for five epochs with the batch size set to 128 and the learning rate set to 2×10^{-5}. The Vicuna-style template is applied to concatenate queries and responses during fine-

Algorithm 1: Complexity-first Diverse Sampling

Data: The Whole Pooled Dataset $\mathcal{D}$, Sub-Dataset Size N
Result: The Sampled Sub-Dataset $\mathcal{D}_s$

```
Initialize Empty D_s;
Sorting Queries in D by tag number in descending;
while |D_s| < N do
    Tag Set T_s^B ← ∅;
    foreach Query q ∈ D do
        if Query Tags T_q : |T_s^B ∪ T_q| > |T_s^B| then
            D_s ← D_s ∪ {q};
            T_s^B ← T_s^B ∪ T_q;
            D ← D \ {q};
            if |D_s| equals to N then
                Break;
return D_s
```

tuning. We evaluate each fine-tuned model on MT-BENCH[3] (Zheng et al., 2023) using GPT4 as a judge to demonstrate the alignment performance, set comparison to other LLMs, and conduct decoupled analyses on data complexity and diversity.

LLMs can benefit more from SFT datasets with higher diversity and complexity according to the analyses in §3.4. We sample an SFT data subset of 6K samples from the pooled dataset with the highest sample complexity of an average tag number 16.56 and tag coverage of 100%. We follow Alg. 1 to obtain the datasets.

4.2 RESULTS

We use the dataset of 6K samples to align the 13B version of LLaMA (Touvron et al., 2023a) and LLaMA-2 (Touvron et al., 2023b) with human preference via SFT, and dub both aligned LLMs TAGLM-13b-v1.0 and TAGLM-13b-v2.0 respectively. We compare our models to two sets of baselines. We first use proprietary GPT-4 (OpenAI, 2023), GPT-3.5 [4], and Claude-V1 [5] as strong state-of-the-art baselines, and then include strong cutting-edge open-resourced aligned LLMs, Vicuna (Chiang et al., 2023), WizardLM (Xu et al., 2023a), Baize (Xu et al., 2023b), OpenChat (Wang et al., 2023a), and Alpaca (Taori et al., 2023). We leave the detailed introductions of these baselines to Appx. §D. For a fair comparison, we all consider the open-resourced LLMs fine-tuned on the same 13B version of LLaMA.

As shown in Tab. 3, we can see that TAGLM-13b-v1.0 outperforms all the open-resourced aligned LLMs achieving a 6.44 average score on MT-BENCH, although it is only fine-tuned based on LLaMA on only 6K samples which amount is far less than those of other LLMs. We report the average of three GPT-4 judgments as we notice there is noticeable randomness in GPT-4 judgments. We also provide the standard deviation of scores in three judgments. This result illustrates that diversity and complexity do matter in improving human alignment performance via SFT. Our INSTAG provides a decent tool for accessing and quantifying both attributes. TAGLM-13b-v2.0 fine-tuned based on LLaMA-2 achieves even higher results while lagging behind LLaMA-2-chat by only 0.1 which is aligned with human preference via RLHF. When compared to proprietary high-performing LLMs, especially GPT-4, the performance is far lagging behind by 2.44 on MT-BENCH.

We also present more detailed scores on MT-BENCH in terms of eight tasks. As shown in Fig. 3, TAGLM-13b-v1 outperforms all other baselines on *stem* and *extraction*, and achieves comparable performances on *humanities* with VICUNA, suggesting these tasks may rely on few data for alignment. TAGLM-13b-v1 ranks the second on *math*, *coding*, and *writing*, but falls short on *roleplay* and

[3] https://huggingface.co/spaces/lmsys/mt-bench

[4] https://platform.openai.com/

[5] https://www.anthropic.com/index/introducing-claude

Model	Data Size	MT-Bench
Proprietary Models		
gpt-4	–	8.99
gpt-3.5-turbo	–	7.94
claude-v1	–	7.90
LLaMA-2 Based Open-source Models		
Llama-2-13b-chat (Touvron et al., 2023b)	–	6.65
TAGLM-13b-v2.0	6K	$6.55_{\pm 0.02}$
LLaMA Based Open-source Models		
alpaca-13b (Taori et al., 2023)	52K	4.53
openchat-13b-v1 (Wang et al., 2023a)	8K	5.22
baize-v2-13b (Xu et al., 2023b)	56K	5.75
vicuna-13b-v1.1 (Chiang et al., 2023)	70K	6.31
wizardlm-13b (Xu et al., 2023a)	70K	6.35
vicuna-13b-v1.3 (Chiang et al., 2023)	125K	6.39
TAGLM-13b-v1.0	6K	$\mathbf{6.44_{\pm 0.04}}$

Table 3: Main results of TAGLM. We present MT-BENCH scores of both proprietary and open-source baselines in similar scales. We use the model name on the Huggingface Hub for all the open-sourced baselines. We also report the approximate data size of each model used in supervised fine-tuning for alignment. We report the average of three GPT-4 judgments and corresponding standard deviations and obtain results for other baselines from the official MT-BENCH leaderboard. Dashes in the data column denote unknown data sizes. Detailed results are presented in Appx. §F.

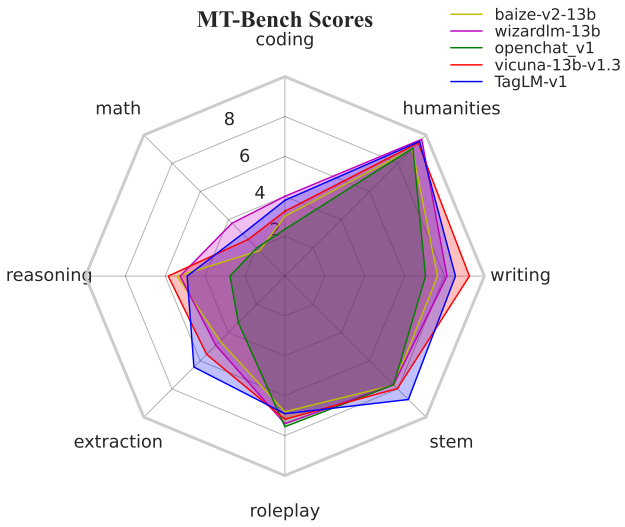
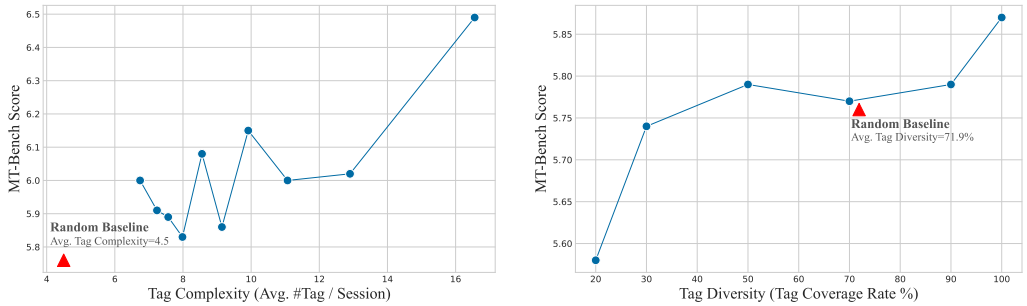


Figure 3: Radar plot showing detailed scores of TAGLM-13b-v1.0 and major baselines on eight subtasks of MT-BENCH. Detailed numbers can be viewed in Tab. 7.

reasoning. These detailed results showing some tasks may require diverse but only a few alignment data, while tasks about reasoning and writing may continually benefit from large-scale data.

4.3 DECOUPLED ANALYSIS

We provide decoupled analyses of complexity and diversity to demonstrate how they influence alignment performance separately.



(a) Performance under different Tag Complexities (b) Performance under different Tag Diversities

Figure 4: Analysis results of model performance in terms of different tag complexities and diversities. Fig. 4a shows MT-BENCH scores over different tag complexities defined as an average number of tags per session. Fig. 4b shows scores over different tag diversities defined as coverage rates over all tags. We include a random baseline in both figures as shown in red triangles.

Complexity. To decouple and focus on the data complexity, we sample different data subsets of diverse averaged tag numbers. Different sampled subsets share the same sample size of 6k and have the same tag coverage of 100% which implies the largest data diversity. In the sampling procedure, all the data samples are first sorted by the tag numbers in descending order. Then for each data subset, we start from the sample in the whole dataset with the largest tag number. The sample that can expand the tag set size of the current sampled data will be extracted and removed from the whole dataset. If the tag set of the current sampled subset covers the whole tag set and the sample number is still less than 6k, we repeat the sampling procedure until the sample numbers reach 6k. This is a similar sampling procedure as Alg. 1. We leave the detailed sampling algorithm for complexity analysis in Appx. §E.

We sample 10 different data subsets, and the average tag numbers of the subsets range from 6.7 to 16.6. As a result shown in Figure 4a, the overall trend of performance on MT-BENCH is increasing along with the growth of average tag numbers. On the fine-grained level of average tag numbers where the number difference between subsets is small, this trend may not be significant. Compared to the randomly sampled datasets the average tag number of which is around 4.5, all the 10 data subsets can lead to superior fine-tuned model performance than the randomly sampled subset baseline. To summarize, on a coarse-grained level of data complexity the downstream performance is positively correlated to the average tag number, while on the fine-grained level, such a phenomenon becomes less evident. This may be partly because CHATGPT does not recall all the possible tags for each query or some tags are filtered out during the tag normalization procedure, resulting in a less accurate tag number for each query.

Diversity. For diversity, we sample different data subsets spanning various tag coverage rates regarding the whole tag set. Different subsets share the same sample scale of 6k and the same average tag number implying the same data complexity. The average tag number is set to 5.0. For data subset sampling, we first draw samples that can expand the tag set size of the current sampled data until the target tag coverage rate. Then we traverse the remaining samples and extract samples that do not expand the tag coverage and can keep the current average tag number of the subset around 5.0. We leave the detailed sampling algorithm for diversity analysis in Appx. §E.

We can observe in Figure 4b that as the tag coverage increases the fine-tuned model can achieve higher MT-BENCH scores. Randomly sampled data subsets of tag coverage 71.9% result in similar model performance with the sampled subset of tag coverage 70%. This demonstrates that through the scope of tags, the fine-tuned models may benefit from the more diverse datasets. The trend is not strictly linear and there seems a plateau ranging from 50% to 90% coverage. This could be due to the tags assigned may not share the sample importance for diversity, for example, the tags *software development* may express more similar semantics with *C++ programming* than those with *information retrieval*.

5 CONCLUSION

In this paper, we introduced INSTAG, an open-set fine-grained tagger that leverages the instruction-following ability of modern chatbots like CHATGPT. Tagging results on open-source SFT datasets show that aligning with more diverse and complex datasets may improve the performance of INSTAG. To explore this insight further, we conducted comprehensive scale analyses on the query diversity and complexity by sampling existing open-source data based on tags from INSTAG. We designed a complexity-first diverse sampling method to sample six thousand queries, and our LLMs fine-tuned on this selected dataset outperformed other open-source models aligned with considerably more data. Moreover, further analyses revealed that model performance increases with more diversity and complexity. In summary, our proposed INSTAG provides a novel aspect for a deeper understanding of query distribution in the alignment of LLMs. It has robust potential to be extended to more applications beyond the data selection shown in this work, such as creating comprehensive evaluations and tag-based self-instruct.

LIMITATIONS

Our conclusions mainly rely on MT-BENCH for model evaluations, which may miss some influence caused by SFT data. Besides, we notice MT-BENCH shows instabilities in terms of the randomness of GPT-4 judgments, so we provide random ablations as comprehensive as possible to show the statistical significance of our results, including reporting standard variance of MT-Bench scores. Furthermore, our analysis of SFT datasets is mainly focused on English, so our claims may not be directly extended to multi-lingual scenarios.

REFERENCES

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a.

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

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

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

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning, 2023.

Michael Hahsler, Matthew Piekenbrock, and Derek Doran. dbscan: Fast density-based clustering with r. *Journal of Statistical Software*, 91:1–30, 2019.

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

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

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations – democratizing large language model alignment, 2023.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. *arXiv preprint arXiv:2012.12624*, 2020.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022. doi: 10.1126/science.abq1158. URL https://www.science.org/doi/abs/10.1126/science.abq1158.

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

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

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

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

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

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

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

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Guan Wang, Sijie Cheng, Qiying Yu, and Changling Liu. OpenChat: Advancing Open-source Language Models with Imperfect Data, 7 2023a. URL `https://github.com/imoneoi/openchat`.

Shufan Wang, Laure Thompson, and Mohit Iyyer. Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. *arXiv preprint arXiv:2109.06304*, 2021.

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

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023c.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023d.

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

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

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

Michihiro Yasunaga and Percy Liang. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning (ICML)*, 2020.

Hongyi Yuan, Keming Lu, and Zheng Yuan. Exploring partial knowledge base inference in biomedical entity linking. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pp. 37–49, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.bionlp-1.3. URL https://aclanthology.org/2023.bionlp-1.3.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

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

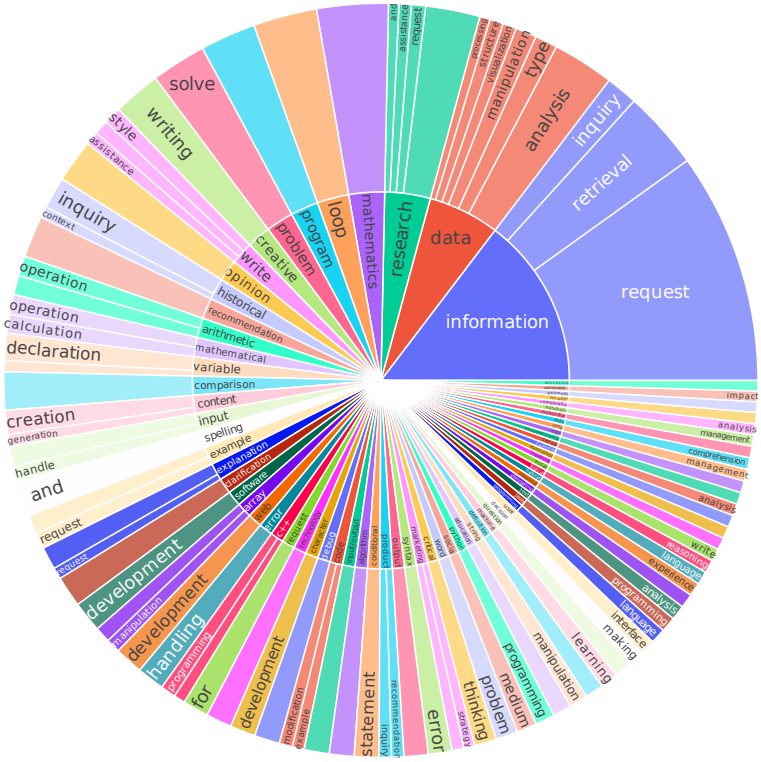


Figure 5: The sunburst plot of all tags. We plot with the first two words of each tag and the size is proportional to the frequency of the tag.

APPENDIX

A TAG REVIEW

We present a sunburst plot of all tags in Fig. 5 showing the most frequent tags is about information-related, data manipulations, and coding queries. We plot with the first two words of each tag and the size is proportional to the frequency of the tag. We only plot with tags that have frequencies larger than 2000 in our data pool.

B PROMPT TEMPLATES FOR CHATGPT

We preset our prompt for CHATGPT for annotation (Tab. 4), precision evaluation (Tab. 5), and consistency evaluation (Tab. 6).

C DATASETS

We apply INSTAG to 17 open-source SFT datasets for intention tagging:

- **ShareGPT**[6] refers to the multi-turn chatting histories used by VICUNA (Chiang et al., 2023). ShareGPT includes human-written queries and responses from CHATGPT and other chatbots.

[6]Exact dataset of ShareGPT (`https://sharegpt.com/`) has not been released. We instead use a reproduced version from `https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered/tree/main/HTML_cleaned_raw_dataset`, and follow Vicuna preprocess.

You are a tagging system that provides useful tags for instruction intentions to distinguish instructions for a helpful AI assistant. Below is an instruction: [begin] {instruction} [end] Please provide coarse-grained tags, such as "Spelling and Grammar Check" and "Cosplay", to identify main intentions of above instruction. Your answer should be a list including titles of tags and a brief explanation of each tag. Your response have to strictly follow this JSON format: [{"tag": str, "explanation": str}]. Please response in English.

Table 4: CHATGPT prompt template for annotating intention tags of given queries.

You are an experienced judge for intention tags of instructions. You will be provided a query and a list of tags describing intentions of the query as followed: [query]: {query} {tags} Please provide feedback about whether all tags precisely describe an intention of the instruction. Please identify all incorrect tags and provide their indices in the JSON format as your response. The JSON format for your response is a list of JSON dictionary and the JSON dictionary has only one key to identify the index of each incorrect tag: [{"idx": int}]. For example, if [tag 0] and [tag 2] are incorrect, you should response as [{"idx": 0}, {"idx", 2}]. If all tags are correct, please response an empty list as [].

Table 5: GPT-4 prompt template for evaluating tagging precision.

- **OpenChat** (Wang et al., 2023a) is a subset of ShareGPT containing only chat histories with GPT-4 responses.[7]
- **UltraChat** (Ding et al., 2023) is a systematically designed, diverse, informative, large-scale dataset of multi-turn instructional conversations without involving human queries.[8]
- **Alpaca** (Taori et al., 2023) is a dataset generated by the modified SELF-INSTRUCT method (Wang et al., 2022), containing 52,000 instruction-following demonstrations generated from OpenAI's *text-davinci-003* model.[9]
- **WizardLM** (Xu et al., 2023a) is an instruction dataset built with the EVOL-INSTRUCT method. EVOL-INSTRUCT utilizes CHATGPT to augment the complexity of the same queries in Alpaca and ShareGPT. We denote these two subsets as WizardLM(Alpaca) and WizardLM(ShareGPT) for clarification.[10]
- **FLAN** (Wei et al.) is a series of data from NLP tasks formatted in instruction tuning. The queries in FLAN are generated by templates for each NLP task.
- **Dolly** (Conover et al., 2023) contains 15,000 high-quality human-generated prompt and response pairs for instruction tuning of LLMs.
- **OAssist** (Köpf et al., 2023) is a crowdsourced human-annotated dataset about multi-lingual conversations.
- **Unnatural** (Honovich et al., 2022) contains queries generated by prompting DAVINCI-002.
- **Lima** (Zhou et al., 2023) contains only 1,000 carefully human-curated prompts and responses.

[7] We use the dataset with 8,000 GPT-4 responses denoting as OpenChat v1.0 in `https://huggingface.co/datasets/openchat/openchat_sharegpt4_dataset`

[8] `https://huggingface.co/datasets/stingning/ultrachat`

[9] We collect the Alpaca dataset along with Dolly, OAssist, and Unnatural from the sharing of Wang et al. (2023b)`https://github.com/allenai/open-instruct`.

[10] We use the V2 version of WizardLM in `https://huggingface.co/datasets/WizardLM/WizardLM_evol_instruct_V2_196k`.

You are an experienced judge for consistency of intention tags for instructions. You will be provided a tag and a list of instructions labeled with this tag as followed: [tag]: {tag} {instructions} Please provide feedback about whether the meaning of this tag is consistent among all instructions. Please identify all inconsistent instructions and provide their indices in the JSON format as your response. The JSON format for your response is a list of JSON dictionary: [{"idx": int}]. For example, if the meaning of tags in [instruction 0] and [instruction 2] are inconsistent, you should response as [{"idx": 0}, {"idx": 2}]. If the meaning of tag is consistent in all instructions, please response an empty list as [].

Table 6: GPT-4 prompt template for evaluating tagging consistency.

- **Math Collections**: We involve a set of math datasets including GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) to prompt INSTAG generating fine-grained mathemetical tags.
- **Code Collections**: We also involve a set of code datasets including DMCC (Li et al., 2022), MBPP (Austin et al., 2021), and DrRepair (Yasunaga & Liang, 2020) for the same purpose as introducing mathematical datasets.

D BASELINE LLMS

We give introductions to the LLM baselines for human alignment.

- **Alpaca** (Taori et al., 2023) is the first open-resourced LLM aligned with human preference. Alpaca is fine-tuned on SFT data of 52K samples generated from text-davince-003 using Self-Instruct (Wang et al., 2023c).
- **WizardLM** (Xu et al., 2023a) is fine-tuned on the SFT data enhanced with a novel technique named Evol-Instruct. It complexifies the Alpaca SFT data using CHATGPT and achieves better alignment performance.
- **Vicuna** (Chiang et al., 2023) is an aligned LLM fine-tuned on collected user chatting logs of proprietary high-performing chatbots on ShareGPT.
- **OpenChat** (Wang et al., 2023a) is fine-tuned on a subset of ShareGPT with only the chatting logs with GPT-4.
- **Baize** (Xu et al., 2023b) uses 100K dialogues generated by self-chatting of CHATGPT. It also includes Alpaca's data for SFT.
- **LLaMA-2 Chat** (Touvron et al., 2023b) differs from the above-mentioned LLMs in (1) being based on per-trained LLaMA-2 instead of LLaMA (Touvron et al., 2023a); (2) being aligned with human preference by both SFT and RLHF.

E SAMPLING ALGORITHM FOR DECOUPLED ANALYSIS

We present our sampling algorithm for decoupled analysis of complexity and diversity in Alg. 2 and Alg. 3, respectively.

F DETAILED RESULTS

We present our detailed results on MT-BENCH in Tab. 7, which is also the source data of Fig. 3.

G COUNTERFACTUAL EVALUATION

To test how well annotators can evaluate tag quality, we created counterfactual cases for two tasks. In the tag precision task, we substituted some tags with similar ones in terms of semantics. In the tag

Algorithm 2: Data Sampling for Complexity Analysis

Data: The Whole Pooled Dataset $\mathcal{D}$, Subset Size N **Result:** The Sampled Sub-Dataset of Different Complexity $\mathrm{D} = \{\mathcal{D}_c^i | i = 1, \ldots, n\}$

```
Sorting Queries in 𝒟 by tag number in descending;
Initialize D = list();
foreach i in {1, ..., n} do
    Initialize Empty 𝒟_c^i;
    while |𝒟_c^i| < N do
        Tag Set 𝒯_c^B ← ∅;
        foreach Query q ∈ 𝒟 do
            if Query Tags 𝒯_q : |𝒯_c^B ∪ 𝒯_q| > |𝒯_c^B| then
                𝒟_c^i ← 𝒟_c^i ∪ {q};
                𝒯_c^B ← 𝒯_c^B ∪ 𝒯_q;
                𝒟 ← 𝒟 \ {q};
                if |𝒟_c^i| = N then
                    D ← D appends 𝒟_c^i;
                    Break;
return D
```

Algorithm 3: Data Sampling for Diversity Analysis

Data: The Whole Pooled Dataset $\mathcal{D}$, Preset Coverage Rate $\mathcal{R} = \{r^i | i = 1, \ldots, n\}$, Subset Size N **Result:** The Sampled Sub-Dataset of Different Diversity $\mathrm{D} = \{\mathcal{D}_d^{r_i} | i = 1, \ldots, n\}$

```
Initialize D = list();
foreach i in {1, ..., n} do
    Initialize Empty 𝒟_d^{r_i} ← ∅;
    Tag Set 𝒯_d ← ∅;
    foreach Query q ∈ 𝒟 do
        if Query Tags 𝒯_q : |𝒯_d ∪ 𝒯_q| > |𝒯_d| then
            𝒟_d^{r_i} ← 𝒟_d^{r_i} ∪ {q};
            𝒯_d ← 𝒯_d ∪ 𝒯_q;
            𝒟 ← 𝒟 \ {q};
            if |𝒯_d|/|𝒯| > r_i then
                Break;
    while |𝒟_d^{r_i}| < N do
        foreach Query q ∈ 𝒟 do
            if Query Tags 𝒯_q : |𝒯_d ∪ 𝒯_q| = |𝒯_d| then
                𝒟_d^{r_i} ← 𝒟_d^{r_i} ∪ {q};
                𝒟 ← 𝒟 \ {q};
                if |𝒟_d^{r_i}| = N then
                    Break;
    D ← D appends 𝒟_d^{r_i};
return D
```

consistency task, we used inconsistent instructions to replace the original instructions. Both humans and GPT-4 are able to recognize most of the counterfactual cases. And humans are better at tag precision, while GPT-4 is better at tag consistency. This analysis shows that annotators have low false positive rates and proof confidence of their judgments in the original tagging results.

Model	Data	MT-BENCH Scores								Average	
		code	extraction	humanities	math	reason	roleplay	stem	writing	all	w/o C&M
gpt-4	–	8.55	9.38	9.95	6.8	9.0	8.9	9.7	9.65	8.99	9.43
gpt-3.5-turbo	–	6.9	8.85	9.55	6.3	5.65	8.4	8.7	9.2	7.94	8.39
claude-v1	–	6.25	8.8	9.7	4.8	5.95	8.5	9.7	9.5	7.9	8.69
Llama-2-13b-chat	-	3.0	6.92	9.75	3.45	5.1	7.5	8.62	8.85	6.65	7.79
TAGLM-13b-v2.0 (1)	6K	3.75	6.5	9.55	2.1	5.3	7.95	8.5	8.75	6.55	7.76
TAGLM-13b-v2.0 (2)	6K	3.7	6.2	9.52	2.15	5.35	8.1	8.4	8.95	6.55	7.75
TAGLM-13b-v2.0 (3)	6K	3.4	7.35	9.6	2.15	5.9	7.45	8.28	8.0	6.52	7.76
INSTAG-v1.0-13b (1)	6K	3.8	6.45	9.55	3.0	4.9	6.9	8.75	8.55	6.49	7.52
INSTAG-v1.0-13b (2)	6K	3.45	6.35	9.65	2.95	4.95	7.15	8.65	8.5	6.46	7.54
INSTAG-v1.0-13b (3)	6K	3.4	6.45	9.45	2.85	5.05	7.05	8.43	8.4	6.38	7.47
vicuna-13b-v1.3	125K	3.25	5.55	9.45	2.6	5.85	7.18	7.98	9.25	6.39	7.54
vicuna-13b-v1.1	70K	2.95	6.4	9.45	2.9	4.65	7.5	8.55	8.05	6.31	7.43
wizardlm-13b	70K	4.0	4.9	9.7	3.75	5.25	7.4	7.7	8.12	6.35	7.18
baize-v2-13b	56K	3.0	4.6	9.02	1.8	5.4	6.8	7.72	7.65	5.75	6.87
nous-hermes-13b	300K	2.45	5.05	9.0	2.65	3.8	6.38	7.02	7.75	5.51	6.5
gpt4all-13b-snoozy	900K	3.0	4.8	8.85	1.2	4.2	7.0	6.9	7.35	5.41	6.52
koala-13b	472K	2.9	4.15	8.45	1.9	4.0	6.85	7.2	7.35	5.35	6.33
openchat-13b-v1	8K	2.35	3.3	9.07	2.0	2.75	7.55	7.7	7.05	5.22	6.24
alpaca-13b	52K	2.35	4.15	7.85	1.05	3.5	5.45	5.2	6.7	4.53	5.48

Table 7: Main results of INSTAG. We present MT-BENCH scores of both proprietary and open-source baselines in similar scales. We also provide average scores overall categories and categories without code and math (w/o C&M). Dashes in the data column denote unknown data scales. Parentheses mark the three different rounds of GPT-4 judgments.

Metric	GPT-4 Annotation		Human Annotation (1%)	
	Original	Counterfactual	Original	Counterfactual
Tag Precision	96.1	6.1	100	0
Tag Consistency	86.6	7.8	100	14.3

Table 8: Evaluation for the tagging quality of INSTAG. We design two metrics, tagging precision and consistency, for evaluating INSTAG. We employ GPT-4 to label 4,000 tagging results. And we also employ three human annotators to annotate 1% cases and report their majority voting. We also create counterfactual cases to probe the judgment abilities of different annotators.

H CASE FROM INSTAG

We present a sample case in Tab. 9 to illustrate the behavior of INSTAG.

I INSTAGGER: LOCAL TAGGER BY DISTILLATION

When it comes to localized deployments or large-scale tagging demands in which cases using CHATGPT or GPT-4 through OpenAI API is not available or infeasible, we may alter to a specialized smaller LLM in the aid of the scenarios. Therefore, we proposed INSTAGGER, which is equipped with the tagging ability of these high-performing chatbots by distillation. Distilling is an effective method to inject a smaller model with specialized abilities, which has been applied to mathematical reasoning abilities recently (Fu et al., 2023). We use our INSTAG results on open-resourced SFT datasets to fine-tune a 7B version LLaMA-2 model. We use a template:

"*You are a helpful assistant. Please identify tags of user intentions in the following user query and provide an explanation for each tag. Please respond in the JSON format {"tag": str, "explanation": str}. Query: <query-to-tag> Assistant: <tagging-results>*"

to concatenate queries to tag and tagging results, and we also include the explanation in the tagging results to make the fine-tuned model obtain better tagging performance. The overall sample size for fine-tuning is 773,511 where we randomly sample 1,000 samples for validation. The model is

Sample Case	
Utterance	
Human:	List 3 breakfast foods that are healthy.
LLM:	1.Oatmeal - it's high in fiber, protein, and antioxidants, and can lower cholesterol levels. 2.Greek yogurt - it's packed with protein, calcium, and probiotics, which support gut health. 3.Eggs - they're rich in protein, vitamins D and B12, and minerals like selenium, and can help you feel full longer."
INSTAG Results	
Tag	**Explanation**
food recommendation	The instruction asks for a list of breakfast foods that are healthy.
quantity	The instruction specifies that the list should include three breakfast foods.
health	The instruction emphasizes the need for the breakfast foods to be healthy.

Table 9: One case from SFT data of WizardLM tagged by INSTAG. INSTAG accurately captures the semantics (food, health) and intentions (recommendation, quantity) in the human query.

fine-tuned with 512 batch size for 1 epoch since we empirically find that training for more than 1 epoch will lead to over-fitting.

We validate the model on our validation set. The tag-level F1 score based on exact match (EM) and semantic-based fuzzy match are 31.8% and 73.4%. As this is an unconstrained open-generated tagging, EM is a very strict metric for annotating over six thousand tags. Therefore, we also calculate the fuzzy match by PHRASEBERT, which considers a predicted tag is correct if it has over 0.8 cosine similarity in semantics with any gold tag.