FANNO: Augmenting High-Quality Instruction Data with Open-Sourced LLMs Only

Anonymous ACL submission

Abstract

Instruction fine-tuning stands as a crucial advancement in leveraging large language models (LLMs) for enhanced task performance. However, the annotation of instruction datasets has traditionally been expensive and laborious, often relying on manual annotations or costly API calls of proprietary LLMs. To address these challenges, we introduce FANNO, a fully autonomous, open-sourced framework that revolutionizes the annotation process without the need for pre-existing annotated data. Utilizing a Mistral-7b-instruct model, FANNO efficiently produces diverse and high-quality datasets through a structured process involving document pre-screening, instruction generation, and response generation. Experiments on Open LLM Leaderboard and AlpacaEval benchmark show that the FANNO can generate high-quality data with diversity and complexity for free, comparable to human-annotated or cleaned datasets like Alpaca-GPT4-Cleaned.

1 Introduction

Large language models (LLMs) have made significant contributions across numerous fields (Zheng et al., 2024a; Wang et al., 2024; Wettig et al., 2024; Fan et al., 2023). Instruction tuning (Ouyang et al., 2022) enhances the model's general capabilities for novel tasks and improves their adherence to directives. However, the development of human-annotated instruction data is prohibitively expensive, and often results in suboptimal outcomes (Srivastava et al., 2022; Conover et al., 2023). This is primarily due to the annotators' cognitive limitations, which hinder achieving a balanced dataset in terms of diversity, complexity, and quality (Srivastava et al., 2022; Conover et al., 2023). Previous works explore the automatic LLM-based annotation of instruction data, with advanced proprietary models (Wang et al., 2022a; Xu et al., 2023) or models trained with seed response-query pairs (Li et al., 2024; Lou et al., 2024). Nevertheless, these approaches often depend on costly APIs (ChatGPT/GPT-4) or require manually crafted seed datasets. Recent studies (Zheng et al., 2024c; Yehudai et al., 2024; Press et al., 2023) aim to construct instruction datasets from scratch; however, the strategies to balance the diversity, complexity, and quality (Liu et al., 2023a) of annotated instruction data are less explored.

Addressing these challenges, we introduce FANNO (Free ANNOtator), a freely accessible framework specifically designed for automatic high-quality instruction annotation. This framework methodically breaks down the annotation process into three distinct phases: document prescreen, instruction generation, and response generation. It utilizes curated tagging, UCB(Upper Confidence Bound) bootstrapping iterations, and filtering techniques to enhance the diversity and complexity of the generated instructions. Empirical evidence on Open LLM Leaderboard and AlpacaEval benchmark confirm the framework's efficacy on two 7B LLMs. The resulting dataset is virtually indistinguishable from those refined datasets like **Alpaca-GPT4-Cleaned**, marking a significant stride in instruction data development[1].

2 Related Work

Instruction Data Generation Two main approaches have been explored for instruction data creation: (1) **Human Annotation**, which leverages human expertise to design prompts and collect multi-task datasets spanning various categories (Srivastava et al., 2022; Conover et al., 2023). While producing high-quality data, manual annotation is effort-intensive and costly, especially for devising complex textual instructions. (2) **LLM Synthetic Data Generation** Recent research increasingly favors harnessing the creative

[1] Our code, data, and model will be made public.

capabilities of LLMs, such as GPT-4 (OpenAI, 2023), over human input for creating instruction-following datasets (Geng et al., 2023; Chiang et al., 2023). ALPACA (Taori et al., 2023) and ALPACAGPT (Peng et al., 2023) have also utilized more powerful LLMs to enhance data quality. Another line of research involves generating task instructions from "seeds" and filtering (Wu et al., 2023). For example, WIZARDLM (Xu et al., 2023) employed an instruction evolution paradigm to increase seed instruction complexity, while SELF-INSTRUCT (Wang et al., 2022a) used human-annotated instructions as demonstrations to guide LLMs in the instruction evolution process. Humpback (Li et al., 2024) generates instructions using vast amounts of unlabeled web text. These datasets are costly, either in terms of labor or proprietary model expenses. In contrast, FANNO maintains high instructional quality autonomously, utilizing open-source models efficiently with just a 7B model size.

Instruction Tuning Instruction tuning involves training LLMs on extensive upstream task datasets with instructions, followed by enabling the generalized ability to new, unseen downstream tasks via new instructions (Ouyang et al., 2022; Chung et al., 2022). This technique is widely acknowledged as essential for activating LLMs to adhere to human conversational norms (Mishra et al., 2022). Instruction tuning has empowered various domain-specific or task-specific LLMs (Jiang et al., 2023b; Xu et al., 2023), and curating diverse, high-quality upstream instruction dataset has become a pivotal step for successful instruction tuning (Wang et al., 2023; Lou et al., 2023). Moreover, instruction tuning also bolsters cross-task general capabilities (Sanh et al., 2022; Wang et al., 2022b), encompassing a more comprehensive array of general tasks, notably incorporating input from users of language models (Ouyang et al., 2022; Peng et al., 2023).

Data Quality Enhancement Related works in the field of enhancing data quality have focused on several key aspects such as instruction difficulty, diversity, and correctness. HUMPBACK (Li et al., 2024) and KUN (Zheng et al., 2024c) utilize language model's capability in combination with tailored prompts for data filtering. In Addition, initiatives like GENIE (Yehudai et al., 2024) and MODS (Du et al., 2023) utilize specialized open-source LLMs for data filtering tasks. DEITA (Liu et al., 2023a), PLANGPT (Zhu et al., 2024) and similar approaches utilize fine-tuned large models to score the data for quality assessment. Moreover, efforts like ORCA-MATH (Mitra et al., 2024) and REFLECTION-TUNING (Li et al., 2023a) employ collaborative approaches with multiple LMs and self-reflection to enhance data quality.

3 FANNO Framework

The FANNO framework aims to annotate diverse, complex, and faithful instruction data with only free open-sourced LLMs. As depicted in Figure 1, FANNO consists of three pivotal steps: document pre-screen, instruction generation, and response generation.

3.1 Document Pre-Screen

The FANNO framework annotates instruction data from web corpus, textbooks, etc. The document pre-screening stage initially includes segmentation, deduplication, and length-based filtering. Further filtering employs a teacher LLM and a fast community detection algorithm to enhance correctness and diversity.

The LLM-based filter addresses ambiguous content, privacy concerns, and advertisements (see Appendix D.1). To reduce data volume while maintaining diversity, we cluster instruction embeddings using a fast community detection algorithm, similar to SentenceTransformer (Reimers and Gurevych, 2019), based on a predefined similarity threshold. This approach prioritizes larger, non-overlapping communities (details in Appendix B.3, Algorithm 1).

The pre-screen phase balances processing speed and precision, prioritizing efficiency. In our experiments, the pre-screen stage filters and keeps 6% of the original raw data.

3.2 Instruction Generation

At this stage, FANNO adopts a bootstrapping approach to generate instructions from pre-screened documents, streamlining the process into two distinct phases: seed instruction generation and instruction augmentation.

Step 1: Seed Instruction Generation This step produces a set of diverse instructions as the initial seeds. Diversity is promoted from two perspectives: **Task Types** and **Difficulty Levels**, for which we have manually created corresponding tags (see Appendix D.4). For each document, we traverse

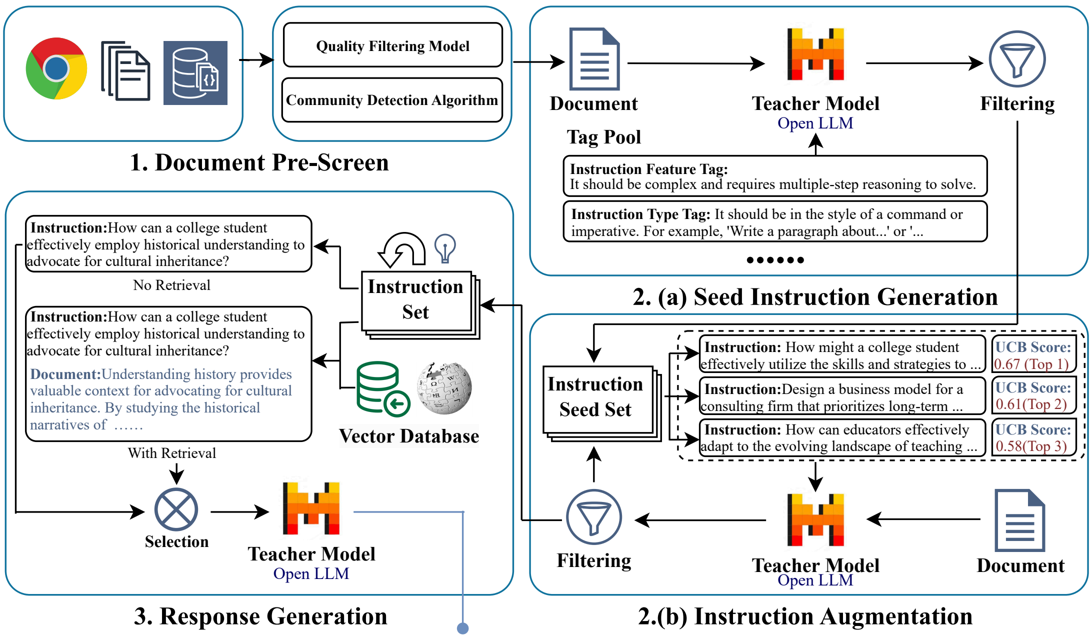

Figure 1: Overview of FANNO framework. **(1) Document Pre-Screen**: We process the unlabeled text data with filters and community detection algorithm. **(2a) Seed Instruction Generation**: FANNO generates seed instructions from pre-screened documents with diverse task types and difficulty levels through a tag pool. **(2b) Instruction Augmentation**: New instructions are augmented conditioned on the documents and few-shot examples selected from the seed instructions with the UCB algorithm. **(3) Response Generation**: The responses to instructions are generated directly by the teacher LLM or based on the concatenation of the corresponding document and retrieved document.

all combinations of task types and instruction difficulty levels to generate seed instructions. An LLM-based filter (see Table 8 in the appendix) is then employed to ensure the quality of the seed instruction data. We sample 200 documents for instruction generation and obtain around 1k instructions as the seed pool S.

Step 2: Instruction Augmentation The diversity of the instructions in S is inherently limited. To promote the diversity of newly generated instructions, we designed a prompt template called **Think Different** (see Appendix 13), which diverges from the traditional example-followed template used in self-instruct (see Appendix D.6). This template encourages the teacher model to generate high-quality instructions that emulate the quality of the examples but differ in format (task types, questioning styles, etc.). Additionally, a document is inputted into this template to ensure the generated instructions are consistent with or extended from this document.

The quality of the examples is, therefore, paramount. Instead of randomly selecting examples, we prioritize extracting higher-quality ones, assuming that instruction length correlates with quality. To avoid suboptimal convergence, the **UCB** (Upper Confidence Bound) (Robbins and Monro, 1951) score is used to enhance the exploration of new instructions. Each seed data is scored as $UCB(s) = \bar{x}_s + C\sqrt{\frac{2\ln N}{n_s}}$. Here, $\bar{x}_s$ is the seed's average quality, N is the total iterations, and C is a constant. The score promotes high-quality and less frequently selected seeds, with C balancing these objectives. In each iteration, we select k seeds with the highest UCB scores, effectively trade-off between exploration and exploitation. We compare UCB and random sampling in an ablation study. The detailed algorithm can be found in Appendix B.2.

3.3 Response Generation

At this stage, the response to each instruction is generated by prompting the teacher LLM either with empty context or a retrieved document. We propose to apply retrieval augmented generation (RAG) and incorporate the corresponding document to provide additional information for response generation. These documents are concatenated to serve as the relevant context. For all generations, the teacher LLM is prompted to generate responses under the above two different conditions. Then, we use the LLM itself to select the response with better quality. The prompt templates in Appendix D.3 and Table 15 are used for response generation and selection, respectively.

3.4 Discussion

To produce diverse and complex high-quality instruction data, FANNO utilizes tags for difficulty balancing, iteratively selects high-quality data via UCB bootstrap, and ensures diversity through iterative instruction filtering. We generated strongly generalized data independent of the original text through carefully crafted prompts. To ensure fidelity between instructions and responses, information is supplemented using RAG and a Teacher model. Detailed discussions are in Section 5.

4 Experiment

4.1 Experiment setup

Unlabeled Text Data We use the FALCON REFINED WEB corpus[2] (Penedo et al., 2023), a large web-based corpus dataset including 600 billion tokens, as our unlabeled data. We directly selected the first 500k documents for input to the Document Pre-Screen stage.

Models and Training Details We choose the Mistral-7b-instruct-v0.2 (Jiang et al., 2023a) for data annotation in all experiments. We perform supervised instruction tuning using LoRA (Hu et al., 2021) with the pretrained LLaMA-2-7b-base model (Touvron et al., 2023) and Mistral-7b-base model (Jiang et al., 2023a). The model after instruction tuning with the data annotated with FANNO framework is referred to as FANNO. Detailed configuration can be viewed in Appendix C.

Baselines We compare FANNO with models fine-tuned with other instruction datasets. The baseline details are in Appendix A. The datasets include Alpaca-52k (Taori et al., 2023), Alpaca-GPT4 (Peng et al., 2023), Alpaca-Cleaned, LIMA (Zhou et al., 2023), WizardLM-70k (Xu et al., 2023), and Muffin (Lou et al., 2024). Alpaca-52k and Alpaca-GPT4, each with 52,002 samples, use Text-Davinci-003 and GPT-4 for annotations. Alpaca-Cleaned refines Alpaca-GPT4 to 51,760 samples filtered instructions with hallucination errors or invalid outputs. LIMA offers 1,000 manually selected diverse prompts and responses. WizardLM-70k and Muffin, both using ChatGPT or GPT-4 annotations, focus on 70,000 and 68,000 high-quality samples, respectively. The self-augmented dataset of Humpback is also comparatively ensured to be fair.

[2] https://huggingface.co/datasets/tiiuae/falcon-refinedweb

4.2 Evaluation

Open LLM Leaderboard The Huggingface Open LLM Leaderboard[3] (Beeching et al., 2023) stands as a unified framework designed to evaluate generative language models across a wide array of diverse evaluation tasks. It encompasses key benchmarks such as ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), and TruthfulQA (Lin et al., 2022). We utilize the lm-evaluation-harness toolkit[4] (Gao et al., 2023) for evaluating different models to maintain consistency with the official setup.

AlpacaEval 2.0 AlpacaEval Benchmark (Li et al., 2023b) is an automated evaluation framework based on a annotation model(GPT-4). By comparing responses generated by two different models for the same set of 805 prompts, AlpacaEval computes the pairwise win rate, automating the evaluation process.

Human Evaluation We employed manual annotation by multiple experts to identify the complexity of instructions, specifically categorized into three tiers: (0 Unanswerable, 1 Easy, 2 Expert). The detailed information for each tier is provided in Appendix F.2. The evaluation results are presented in Table 5 and discussed in Section 5.2.

MT-Bench The MT-Bench (Multi-turn Benchmark) (Zheng et al., 2024b) is aimed at assessing the conversational and instruction-following abilities of LLMs. It comprises 80 multi-turn questions, and GPT-4 is utilized as an automated evaluator, scoring chatbot responses on a scale of 1 to 10, with methods in place to minimize bias and enhance the reliability of the assessments.

4.3 Results

The comparative experiments of FANNO with other models demonstrate the superiority of our work. (1) For diverse base models like LLaMA and

[3] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

[4] https://github.com/EleutherAI/lm-evaluation-harness

Model	Data Size	ARC	HellaSwag	MMLU	TruthfulQA	Average
			Open-sourced Models based on LLaMA-2			
LLaMA-2-Base	–	54.10	78.71	45.80	38.96	50.76
LLaMA-2-Chat	–	54.10	78.65	45.69	44.59	55.76
LLaMA-2 + Alpaca-52k	52k	54.78	78.17	46.65	41.43	55.26
LLaMA-2 + Alpaca-GPT4	52k	56.66	78.78	46.96	51.02	58.35
LLaMA-2 + Alpaca-Cleaned	51.8k	56.40	80.16	47.02	50.53	58.53
LLaMA-2 + LIMA	1k	54.61	79.21	45.79	41.32	55.23
LLaMA-2 + WizardLM-70k	65k	54.01	78.66	45.61	38.99	54.32
LLaMA-2 + Muffin	68k	54.10	76.97	47.12	43.51	55.42
LLaMA-2 + FANNO	16k	55.63	79.45	46.84	51.01	58.23
			Open-sourced Models based on Mistral-7B			
Mistral-7B-Instruct-v0.2	–	59.39	84.33	59.28	66.79	67.45
Mistral-7B-Base-v0.1	–	60.84	83.31	62.42	42.59	62.29
Mistral-7B-Base + Alpaca-GPT4	52k	63.65	82.18	59.29	43.98	62.29
Mistral-7B-Base + Alpaca-Cleaned	51.8K	64.51	83.68	59.76	52.00	64.99
Mistral-7B-Base + FANNO	16k	64.16	85.08	60.79	52.16	65.55

Table 1: Open LLM Leaderboard results evaluated with the lm-evalution-harness toolkit. Data size represents the number of samples in the instruction data.

Table 2: Comparison of Different Models on MT Bench

Model	MT Bench
LLaMA-2-7B	3.97
LLaMA-2-7B (alpaca-gpt4)	4.31
LLaMA-2-7B (self-instruct–Teacher: Mistral)	4.96
LLaMA-2-7B-chat Official implementation	6.27
LLaMA-2-7B (FANNO–Teacher: LLaMA-2-Chat)	4.65
LLaMA-2-7B (FANNO–Teacher: Mistral)	5.11

Mistral, our framework consistently achieves top rankings in the LLM-open-leaderboard, even rivaling the models fine-tuned with Alpaca-GPT4-Cleaned, which underwent augmentation with proprietary models and manual selection. (2) Compared to other similar automatic instruction annotation frameworks like Humpback, Muffin, WizardLM, we adhered to the principle of fairness as much as possible by fully utilizing the officially published datasets, and experiments proved that FANNO achieved excellent results with a smaller dataset.

Using MT-Bench in Table 2, we observe that our model outperforms those fine-tuned using Alpaca-GPT4-clean, highlighting the effectiveness of FANNO. Moreover, compared with *self-instruct*, we relieve the need for manually labeled data and achieve better performance, while naive *self-instruct* with Mistral does not yield optimal results. However, it is understandably inferior to the LLaMA2-7B-Chat model, which benefits from extensive fine-tuning and RLHF alignment.

Models refined through FANNO exhibit notable enhancements in the TruthfulQA metric and show measurable improvements across three other metrics. This advancement is attributed to the integration of supplementary information via the RAG component and self-reflective teacher model, thereby improving the model's proficiency in delivering more faithful outputs and bolstering TruthfulQA scores. Slight improvements in ARC, HellaSwag, MMLU metrics are credited to the elevated challenge and diversity of the instructions, as depicted in Table 1. We also uploaded our model to Huggingface Open LLM Leaderboard and compared our results with models like Vicuna, and Humpback, which are shown in Table 3. As shown in Figure 2, our model marginally outperforms the Alpaca-GPT4-Cleaned's fine-tuned variant on the AlpacaEval benchmark, attesting to the superiority of our FANNO framework.

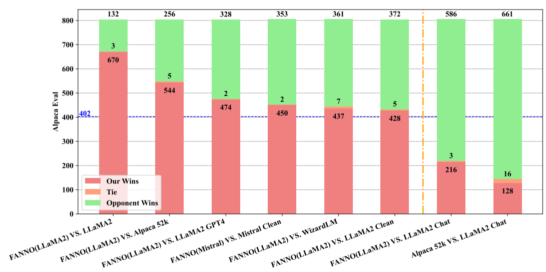

Figure 2: AlpacaEval Result

4.4 Ablation Study

To enhance our understanding of the functionality of each module within FANNO, we undertook ablation studies on its four components, as delineated in Table 4. Our findings reveal:

Model	ARC	HellaSwag	MMLU	TruthfulQA	Average
LLaMA-2-Base	53.07	78.59	46.87	38.76	54.32
LLaMA-2-Chat	52.90	78.55	48.32	45.57	56.34
Vicuna-7b-v1.3	50.43	76.92	48.14	47.01	55.62
Humback M_0	56.31	81.20	47.45	47.59	58.13
Humback M_1	52.99	78.57	45.48	41.45	54.65
WizardLM-7b	54.95	77.85	45.79	48.29	56.72
FANNO	55.46	79.29	46.58	52.05	58.35

Table 3: Benchmark results evaluated by the official Huggingface Open LLM Leaderboard platform.

ID	Configuration	Open LLM Leaderboard				Average
		ARC	HellaSwag	MMLU	TruthfulQA	
① 0	Base	54.44	78.66	44.69	46.02	55.95
① 1	Pre-Screen	55.46	78.51	46.00	45.85	56.44
② 2	FANNO (w/o Iter and UCB)	54.69	79.18	45.92	50.19	57.50
② 3	FANNO (w/o UCB)	55.63	79.43	44.84	51.16	57.77
③ 4	FANNO (w/ OD)	55.46	78.31	44.99	45.68	56.11
③ 5	FANNO (w/ RAG)	55.29	78.41	45.80	45.37	56.22
③ 6	FANNO (w/ RAG+)	55.03	78.46	47.02	46.26	56.69
③ 7	FANNO	55.63	79.45	46.84	51.01	58.23

Table 4: Ablation results from the lm-evaluation-harness(Gao et al., 2023). (0). Basic framework: simply generate instructions by documents and generate responses by instructions without any optimization. (1). Add Pre-Screen module into the basic framework. (2). FANNO without Iteration and UCB-selection. (3). FANNO without UCB-selection. (4). FANNO with the original document. (5). FANNO with RAG module. (6). FANNO with RAG module and supplementary materials. (7). The complete version of FANNO.

- **Orthogonality of Components and Separate Optimization** We replaced each component with a random strategy, and the experiments show that each module positively affects the model's performance, and using more advanced strategies yields better results. ① indicated that the Pre-Screen strategy helps to enhance the quality and thematic diversity of the raw documents. Configurations (3) and (7) demonstrated that using the UCB strategy in instruction augmentation balances the complexity and diversity of the generations, achieving higher diversity compared to random sampling. The notable growth in the MMLU result, as indicated by the ② & 7 combinations, revealed that iterative enhancements in conjunction with the UCB strategy were paramount. The UCB's proactive selection of high-quality data for augmentation facilitates a gradual evolution towards more effective methodologies as the iteration progresses.
- **Generalizing Boosting Diversity and Complexity** As introduced in Section 3, FANNO uses randomness, deduplication, and carefully designed prompts to increase the diversity of themes and tasks in instructions as much as possible. In this way, FANNO tends to break away from reliance on the corresponding unlabeled documents, enhancing the generalizability of instructions. Ablation experiments ① & 7 proved that texts with higher generalizability exhibit more diversity and complexity, which is more beneficial for activating the capabilities of the base models.
- **Knowledge Supplementation Promotes Instruction Quality** The results of ③ demonstrated that it is necessary to incorporate RAG or supplement knowledge with the help of a teacher model is necessary. We discovered that considering only the direct generation of the teacher model yielded the best results compared to the document-based response, particularly on TruthfulQA. This indicates that instructions generated with the FANNO framework are more general and less reliable on the corresponding document. We used RAG in an experiment to pinpoint the most relevant content for the instruction, and experiments 5 & 6 support our assertion that more is better. Other discussions about the truthfulness are covered in Section 5. RAG+ used larger datasets than RAG, but both were from Wikipedia.[5]

5 Analyses

In this section, we will discuss how diversity, correctness, and complexity are promoted in each

[5]The RAG used 2.73GB of data from Wikipedia's introduction section, and the RAG+ used 20.28GB of data from Wikipedia.

stage.

5.1 Analyses of the Augmented Instruction Data

We analyze and illustrate the generated instructions of our dataset from 4 aspects:

Length To study the distribution of the length of instructions, we tokenize each instruction combined with input and count the words within it as its length. Figure 5 and Figure 7 in Appendix E.1 illustrate the distribution of instruction length for FANNO and Alpaca-Cleaned, respectively. The results show that FANNO instructions are more balanced than Alpaca-Cleaned and the mean value of lengths is higher than that of Alpaca, which indicates a better performance.

Diversity Inspired by SELF-INSTRUCT (Wang et al., 2022a), the verb-noun pairs in instructions to represent the types and tasks of instructions are identified and extracted, which exhibits diversity. As Figure 6 and Figure 8 in Appendix E.1 depicted, FANNO instructions possess more challenging verb-noun pairs than Alpaca-Cleaned, which indicates more challenging tasks. The extraction is completed by Berkeley Neural Parser (Kitaev and Klein, 2018; Kitaev et al., 2019).

Quality and Complexity To evaluate the quality and complexity of instruction-response pairs, we utilize Deita-quality-scorer model and Deita-complexity-scorer model (Liu et al., 2023a) as an evaluator to score our instructions. Figure 9 in Appendix E.1 shows the quality and complexity comparison between FANNO and Alpaca-Cleaned, of which the result shows that FANNO instructions possess a more balanced complexity distribution and higher average quality. The corresponding prompts can be found in Table 9 and Table 10 in the appendix.

5.2 Randomness Tag Boosting the Complexity

Score	Tagged	Untagged
0 (bad)	18	24
1 (everyday)	39	78
2 (expert)	143	98

Table 5: Randomness Tag Evaluation Results

Randomness tags serve as additional requirements for the teacher model when generating instructions, enhancing their complexity. To demonstrate the effectiveness of this strategy, generated instructions are manually annotated to evaluate their complexity, as discussed in Section 4.2. We randomly sampled 200 instances from two datasets for manual testing: one utilizing the Random tag strategy (Tagged) and the other generated directly (Untagged). Results are summarized in Table 5, with example instructions detailed in Appendix F.2. From the results, it is evident that instructions with random tags exhibit a significant increase in complexity, manifested by a greater number of expert-level instructions and fewer daily instructions. It is worth noting that instructions with random tags exhibit a tendency towards greater length, including few overly complex tasks that are difficult to answer (classified as expert-level instructions), which indicates a potential need for further refinement.

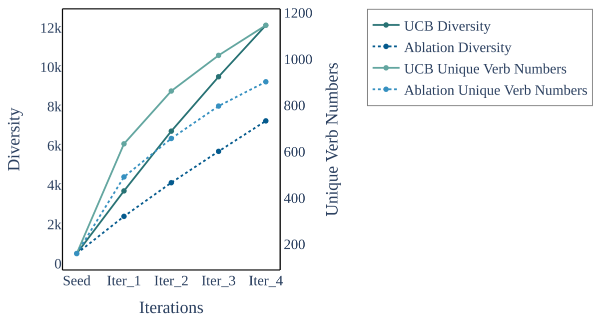

Figure 3: The verbs-noun statistics data grows with iteration

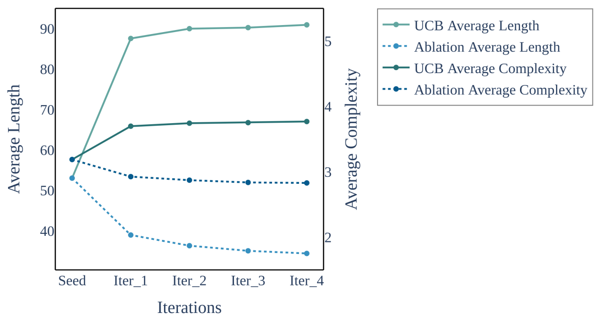

Figure 4: The instruction length (complexity) grows with iteration

5.3 UCB Bootstrap Iteration Improve Instruction Complexity while Maintaining the Diversity

UCB Bootstrap is employed to actively stabilize the process of instruction improvement. For illustration, we monitored the diversity and complexity of instructions during iterations and compared it with a random selection strategy. Note that to simplify the process, we use the length of instructions in words as the measure of quality. As depicted in Figure 3 and Figure 4, we observed an increase in

		Open LLM Leaderboard				Average
		ARC	HellaSwag	MMLU	TruthfulQA	
direct-based	Fearless Responses	54.86	79.33	45.56	47.40	56.79
	Cautious Response	54.44	79.17	45.88	47.29	56.69
doc-based	Faithful Responses	55.03	78.63	45.66	42.82	55.54
	Adaptive Responses	55.63	78.71	45.86	42.76	55.74

Table 6: Comparison results of four types response on the Open LLM

both the average complexity and diversity scores as the iteration progressed, consistent with our expectations. We analyze that UCB prioritizes exploring longer instructions for few-shot instruction generation, resulting in more challenging instructions. Additionally, UCB exhibits a preference for selecting newly generated instructions, as novel few-shot combinations tend to ignite the model's creativity.

5.4 Truthfulness is Less Important for Capability Activation

Previous work (Zhou et al., 2023; Liu et al., 2023b) has explored the diversity, complexity, and fidelity of instructions that enhance large models' capabilities. We further investigate the truthfulness of responses to instructions, noting that responses often seem accurate but contain illusions and fabricated information, potentially affecting instruction fine-tuning.

To address this, we selected about 1,000 expert-level instructions from the FANNO dataset, then prompted LLM to generate four different responses for each instruction with the following settings:

- Fearless Response: Models provide answers regardless of correctness.
- Cautious Response: Models may acknowledge a lack of knowledge.
- Faithful Response: Models generate answers solely based on provided documents.
- Adaptive Response: Models use relevant information from provided documents to generate answers.

From the result in Table 6, an intriguing finding is that direct responses from the model, which contains a substantial presence of illusions, outperformed document-based ones, particularly in the TruthfulQA task. This might suggest that providing human-like and consistent responses, even with false data, can also improve the model's capabilities during SFT. We also need to point out FANNO also introduces external sources of information such as the knowledge of the teacher model itself, which likewise results in some illusory responses.

6 Limitations

While FANNO has demonstrated outstanding performance, several limitations must be acknowledged. The responses are not entirely dependent on the document, leading to the introduction of certain hallucinations in the fine-tuning data, as discussed in Section 5.4. This suggests that the model's reliance on the provided context needs to be strengthened to improve factual consistency. The simplistic approach of equating instruction length with its value is rather crude. The true value of an instruction is influenced by various factors such as difficulty, quality, and novelty. Future work will aim to develop a more nuanced understanding and evaluation of instruction value. The quality of generated instructions is contingent upon the capabilities of both the generator and the evaluator. This process is sensitive to the teacher model and the prompts used, indicating a need for designing prompts that are specifically tailored to the model. Addressing these limitations will be a focus of our future work.

7 Conclusion

The development of instruction fine-tuning datasets has been hindered by the high cost and labor-intensive nature. In this paper, we introduced FANNO, an autonomous and low-cost framework that addresses these challenges by streamlining the annotation process with open-sourced LLMs. FANNO efficiently generates datasets of high quality, diversity, and complexity through a structured process involving pre-screening, instruction generation, and response generation. This unified process eliminates the need for pre-existing annotated data or costly API calls, marking a significant advancement in instruction data development. Empirical experiments also validate the efficacy of FANNO, underscoring the framework's potential to democratize access to high-quality instruction datasets. FANNO enables access to top-quality datasets with reduced cost and effort, driving progress in LLM applications.

References

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling Instruction-Finetuned Language Models. *arXiv preprint arXiv:2210.11416*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Mike Conover, Matt Hayes, Matt Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Ali Ghodsi, Patrick Wendell, and Patrick Zaharia. 2023. Hello dolly: Democratizing the magic of chatgpt with open models.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *Preprint*, arXiv:2311.15653.

Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M. Zhang. 2023. Large language models for software engineering: Survey and open problems. *Preprint*, arXiv:2310.03533.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2023b. Tigerscore: Towards building explainable metric for all text generation tasks. *Preprint*, arXiv:2310.00752.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

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

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Heng Huang, Jiuxiang Gu, and Tianyi Zhou. 2023a. Reflection-tuning: Data recycling improves llm instruction-tuning. *ArXiv*, abs/2310.11716.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024. Self-alignment with instruction backtranslation. *Preprint*, arXiv:2308.06259.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *Preprint*, arXiv:2312.15685.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023b. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *Preprint*, arXiv:2312.15685.

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

Renze Lou, Kai Zhang, and Wenpeng Yin. 2023. Is prompt all you need? no. a comprehensive and broader view of instruction learning. *arXiv preprint arXiv:2303.10475*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *Preprint*, arXiv:2402.14830.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *Preprint*, arXiv:2306.01116.

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

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. *Preprint*, arXiv:2210.03350.

Sebastian Raschka. 2023. Finetuning llms with lora and qlora: Insights from hundreds of experiments. *Lightning AI*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

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

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

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024. Can llms reason with rules? logic scaffolding for stress-testing and improving llms. *Preprint*, arXiv:2402.11442.

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

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

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

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. *Preprint*, arXiv:2402.09739.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023.

Lamini-lm: A diverse herd of distilled models from large-scale instructions. *arXiv preprint arXiv:2304.14402*.

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

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *Preprint*, arXiv:2401.14367.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *Preprint*, arXiv:1905.07830.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024a. Take a step back: Evoking reasoning via abstraction in large language models. *Preprint*, arXiv:2310.06117.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Tianyu Zheng, Shuyue Guo, Xingwei Qu, Jiawei Guo, Weixu Zhang, Xinrun Du, Chenghua Lin, Wenhao Huang, Wenhu Chen, Jie Fu, et al. 2024c. Kun: Answer polishment for chinese self-alignment with instruction back-translation. *arXiv preprint arXiv:2401.06477*.

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. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

He Zhu, Wenjia Zhang, Nuoxian Huang, Boyang Li, Luyao Niu, Zipei Fan, Tianle Lun, Yicheng Tao, Junyou Su, Zhaoya Gong, Chenyu Fang, and Xing Liu. 2024. Plangpt: Enhancing urban planning with tailored language model and efficient retrieval. *Preprint*, arXiv:2402.19273.

A Experiment Baselines

- Alpaca-52k (Taori et al., 2023). This dataset is developed by Stanford University using Text-Davinci-003. It encompasses 52,002 instruction-following samples.

- Alpaca-GPT4 (Peng et al., 2023). This dataset contains English Instruction-Following Data generated by GPT-4 using Alpaca prompts for fine-tuning LLMs. It encompasses 52,002 instruction-following samples, the same as Alpaca-52k.

- Alpaca-Cleaned. This is a cleaned version of the Alpaca-GPT4 Dataset to address problems like hallucinations, merged instruction, and so on. It encompasses 51,760 instruction-following samples.

- LIMA (Zhou et al., 2023).This is a dataset of 1,000 prompts and responses from a variety of sources, primarily split into community Q&A forums and manually authored examples, where the outputs (responses) are stylistically aligned with each other, but the inputs (prompts) are diverse.

- WizardLM-70k (Xu et al., 2023). This dataset employs the Evol-Instruct algorithm to enhance the quality of instruction data. Incorporating ChatGPT during the reformulation phase ensures the data fidelity. Among its 250,000 instructions, we primarily focused on the WizardLM-7b subset, which consists of 70,000 samples.

- Muffin (Lou et al., 2024) MUFFIN's data curation includes input sampling, instruction collection via two methods, output annotation by ChatGPT/GP4-4, instruction filtering, and classification expansion. This is a large dataset of 68k training instances.

- ShareGPT (Chiang et al., 2023). This is a human-annotated dataset consisting of approximately 70K user-shared conversations collected from ShareGPT.

- Humpback. This self-alignment method generates instruction data through reverse fine-tuning.

B FANNO Details

B.1 Pre-screen Details

Our objective was to efficiently enhance the selection process, minimizing time spent while maximizing quality outcomes. Initially, we employed **Mistral-7b-instruct-v2** (Jiang et al., 2023a) to evaluate texts for repetitive content, personal privacy concerns, specific themes, and advertising, using prompts to guide scoring and annotation (see Table 7). For diversity assessment, we utilized a fast community detection algorithm 1 with hyperparameters set to k = 2 and simratio = 0.7(k: the minimum size of a community; simratio: controls the similarity threshold, Only node pairs with similarity scores higher than this threshold are considered connected), facilitating the classification of half a million entries within minutes. The model paraphrase-MiniLM-L6-v2 (Reimers and Gurevych, 2019)) is used for text embedding. For larger datasets, texts were segmented into groups for individual community detection analyses. After the pre-screening process, *Pre-Screen Data* has approximately 30k records, which is 6% of the original. This stage was designed to balance the trade-off between processing speed and analytical precision, prioritizing efficiency over exhaustive detail examination.

B.2 UCB Bootstrap

The setup comprises a language model G parameterized by θ_G for generating instructions, a critic model J parameterized by θ_J for evaluating instruction quality, as well as a document set $\mathcal{D}$, a subset $\mathcal{D}'$, task-type tags $\mathcal{T}_T$, and difficulty-level tags $\mathcal{T}_\mathcal{D}$.

The procedure is as follows:

1. **Initialization:**

$$S \leftarrow \emptyset$$

2. **Seed Generation** ($SeedGen$)**:**

$$\forall d \in \mathcal{D}^{'}, \, generate \, x_i \sim P(x|d, t; \theta_G)$$
$$where \, t \sim \mathcal{U}(\mathcal{T}_{\mathcal{T}} \times \mathcal{T}_{\mathcal{D}})$$
$$S \leftarrow S \cup \{x_i\}$$

3. **Instruction Augmentation** ($InsAug$)**:** For f rounds or until $|S|$ reaches a desired threshold:

   a. Select a subset $S' \subset S$ using the UCB strategy:

$$UCB(s) = \bar{x}_s + C\sqrt{\frac{2 \ln N}{n_s}}$$
$$S' = \{s_i | s_i \in S, UCB(s_i) \text{ is maximized}\}$$

   where $\bar{x}_s$ is the average quality score of instruction s, N is the total number of selections, C is a hyper-parameter constant used to control exploration, t and n_s is the number of times instruction s has been selected.

   b. For each $s_i \in S'$, generate augmented instructions x':

$$x' \sim P(x|c, S'; \theta_G)$$
$$\text{s.t. } Sim(x'; s_i) < \tau$$

   where τ is a similarity threshold.

   c. Update S with the augmented instructions:

$$S \leftarrow S \cup \{x'\}$$

B.3 Fast Community Detection Algorithm

As Algorithm 1 has shown, the Fast Community Detection Algorithm is used to cluster the embeddings of instructions processed by SentenceTransformer (Reimers and Gurevych, 2019), which can then represent the diversity of instructions. Specifically, Fast Community Detection works by iteratively identifying groups of data points (embeddings of sentences) that are closely related based on a predefined similarity threshold, efficiently leveraging cosine similarity calculations. It prioritizes larger communities while minimizing overlapping clusters to produce meaningful community structures.

C Experiment Setting Detail

We chose LoRA over full fine-tuning due to similar performance observed in preliminary experiments, with computational constraints being the primary factor influencing this decision.

We use the same hyperparameters as existing supervised instruction tuning methods (Chiang et al., 2023; Raschka, 2023). Specifically, we use cosine learning rate scheduling with a starting learning rate of 2×10^{-5} and a weight decay of 0.1. The batch size is 32 and the dropout rate is 0.1. For the LoRA configuration, we employ a rank of 256 and set α to 512, with an initial learning rate of 5×10^{-5}. We utilize 8 NVIDIA 4090 GPUs to train our model.

D Prompt Templates Used in FANNO

D.1 Text Filtering

Algorithm 1 Fast Community Detection (Reimers and Gurevych, 2019)

```
function CommunityDetection(embeddings, threshold, min_community_size, batch_size)
    Normalize embeddings
    Initialize extracted_communities as empty list
    for start_idx in range(0, length(embeddings), batch_size) do
        Compute cosine similarity scores for batch starting from start_idx
        Find top-k values from cosine similarity scores
        for i in range(length(top_k_values)) do
            if last element of i-th top-k values ≥ threshold then
                Find top-k most similar entries for i-th element
                while last element of top-k values > threshold and sort_max_size < length of embeddings do
                    Increase sort_max_size if needed
                end while
                Add indices of entries with similarity ≥ threshold to extracted_communities
            end if
        end for
    end for
    Sort extracted_communities by size
    Remove overlapping communities from extracted_communities
    return extracted_communities
end function
```

Table 7: Prompts for Pre-Screen

```
You are act as a assistant to check useless, informal or ambiguous information. Let's think step by step.
The objective is to meticulously inspect the text to determine if it is useless, informal or ambiguous text (e.g. random characters, ambiguous paragraph, broken sequence, informally organized text, etc.)
Your response should be '1' (yes) if the text contains useless, informal or ambiguous information, or '0' (no) if it does not, without providing any reasoning and explanation.

### Document:
{doc}

### Answer:
```

```
You are act as a assistant to check privacy information. Let's think step by step.
The objective is to meticulously inspect the text to determine if it contains any privacy information (e.g. human names, phone numbers, addresses, etc.).
Your response should be '1' (yes) if the text contains privacy information, or '0' (no) if it does not, without providing any reasoning and explanation.

### Text:
{doc}

### Answer:
```

```
I want you to act as an advertisement evaluator. Let's think step by step.
The objective is to meticulously inspect the text based on certain characteristics and decide whether it is an advertisement or not.
Your response should be '1' (yes) if the text is an advertisement, or '0' (no) if it is not, without providing any reasoning and explanation.

Evaluate the text considering these characteristics:

- Promotional language or sales pitch
- Mention of product or service benefits
- Call to action (e.g., "Buy now", "Subscribe")
- Pricing information or special offers
- Contact information or links for more details

<Answer Format>: 1 or 0

### Text:
{text}

### Answer:
```

Table 8: Prompts for instruction generation filter

```
I want you to act as an instruction evaluator. Please evaluate this instruction and respond with
'0' (bad) or '1' (good), without giving reasons.
Standard: A good instruction Must not involve recent or current events. Historical events are
fine.
Example1:
Instruction: Please analyze the recent COVID-19 outbreak.
Answer: 0 (Reason: recent)
Example2:
Instruction: What's happening in China in September 2023?
Answer: 0 (Reason: in September 2023)
Example3:
Instruction: Provide an account of events from last Monday night.
Answer: 0 (Reason: last Monday night)

### Instruction:
{instruction}
### Answer:
```

```
I want you to act as a instruction evaluator. Please evaluate this instruction and respond with
'0' (bad) or '1' (good), without giving reasons.
Standard: A good instruction must not include any private information like names, addresses, phone
numbers, etc, unless the person is historical or famous.
Example1:
Instruction: What is the name of the person who lives at 123 Main Street?
Answer: 0 (Reason: private information)
Example2:
Instruction: What is the name of the first president of the United States?
Answer: 1 (Reason: historical)
Example3:
Instruction: What is the address of the CEO of Microsoft?
Answer: 0 (Reason: private information)

### Instruction:
{instruction}
### Answer:
```

```
I want you to act as a instruction evaluator. Please evaluate this instruction and respond with
'0' (bad) or '1' (good), without giving reasons.
Standard: A good instruction is perfectly logical, and practical, and can be fully understood by a
human.
A bad instruction, likely generated by AI, is generally vague, weird, complex, and long. It may
seem to string unrelated words, topics, and tasks together.
Example1:
Instruction: Considering the health benefits of a non-dairy diet, how does the emotional response
of individuals vary when they attend social events where dairy-based foods are served?
Answer: 0
Example2:
Instruction: Create a multidisciplinary essay that explores the and historical origins of the
dish 'Shrimp Alfredo Pasta Bake'. Discuss the various ingredients, their origins. Additionally,
translate the recipe instructions from English to Spanish.
Answer: 0

### Instruction:
{instruction}
### Answer:
```

Table 7 shows the prompts for basic filtering, including filtering information with useless information, privacy information, or advertisement.

Table 8 shows the prompts for instruction generation filtering, including filtering instructions that are time-sensitive, asking for private information, or not answerable.

D.2 Complexity and Quality Scorer

Table 9: Prompt for quality scorer

```
You are a helpful assistant. Please identify the quality score of the Response corresponding to
the Question.
### Question:
{instruction}
### Response:
{output}
### Quality:
```

Table 10: Prompt for complexity scorer

```
You are a helpful assistant. Please identify the complexity score of the following user query.
### Query:
{instruction}
### Complexity:
```

As Table 9 and 10 have shown, the prompts are provided to deita-complexity-scorer and deita-quality-scorer model (Liu et al., 2023a).

D.3 Generating Instruction Pairs

FANNO employs 2 ways to generate instruction response:

- Question, Document to Answer: model infers answer with both question and related document.
- Question to Answer: model infers the answer directly with the question, using its own knowledge.

Table 11: Question, Document to Answer

```
You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while
being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous,
or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
If a question does not make any sense, or is not factually coherent, explain why instead of
answering something not correct. If you don't know the answer to a question, please don't share
false information.
### Instruction: {question}.
### Paragraph: {doc}.
### Response:
```

D.4 Seed Generation

Listing 1: Seed Generation

```python
def seed_gen(text):
```

Table 12: Question to Answer

```
You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature. If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.
### QUESTION: {question}
### Response:
```

```
    reasoning_tag = "It should be complex and requires multiple-step reasoning to solve."
    critical_thinking_tag = "It demands critical thinking skills to analyze from various perspectives and evaluate multiple solutions."
    creativity_tag = "It necessitates creative thinking to devise innovative solutions beyond conventional approaches."
    interdisciplinary_tag = "It demands integrating knowledge from diverse disciplines to address its multifaceted nature."
    command_tag = "It should be in the style of a command or imperative. For example, 'Write a paragraph about...' or 'Describe the...'"
    question_tag = "It should be in the style of a question or interrogative. For example, 'What is the...?' or 'How do you...?'"

    nli_tag = "It is a Natural language inference question: Assessing if evidence supports a conclusion."
    commonsense_tag = "It is a Commonsense question: Predicting outcomes based on everyday knowledge."
    sentiment_tag = "It is a Sentiment analysis question: Determining emotional response to a given scenario."
    paraphrase_tag = "It is a Paraphrasing question: Rewording a statement while retaining its meaning."
    close_book_qa_tag = "It is a Close-book QA question: Answering factual queries using pre-existing knowledge."
    struc2text_tag = "It is a Structure to text question: Describing a process or concept in written form."
    summarization_tag = "It is a Summarization question: Condensing key information from a larger text."
    translate_tag = "It is a Translation question: Converting text from one language to another."
    implicit_reasoning_tag = "It is a Implicit reasoning question: Inferring reasons behind common behaviors."
    text_category_tag = "It is a Text categorization question: Identifying defining characteristics of a given text type."

    tags = [reasoning_tag, critical_thinking_tag, creativity_tag, interdisciplinary_tag]
    classify = [nli_tag, commonsense_tag, sentiment_tag, paraphrase_tag, close_book_qa_tag, struc2text_tag, summarization_tag, translate_tag, implicit_reasoning_tag, text_category_tag]
    types = [command_tag, question_tag]

    QUESTION_TEMPLATE = """You're proficient in crafting complex question. Generate only one question that adheres to the provided #Paragraph#.
    The question should meet the following criteria:
    0. The person answering the question cannot see the #Paragraph#[SYSTEM: IMPORTANT], so the question must not contain phrases like 'Given the information provided', 'Based on the provided information', or similar expressions that imply direct citations or references from #Paragraph#.
    1. {characteristic}.
    2. {type}.
    3. {classify}.

    ### Paragraph:
    {text}
```

```
    ### Question:
    """
    prompts = [QUESTION_TEMPLATE.format(characteristic=tag, type=type, text=text,
        classify=c) for tag in tags for c in classify for type in types]
    return prompts
```

Code 1 shows the process of generating seed with sampled tags, including task types and difficulty levels.

D.5 Think Different Prompt

Table 13: Prompt for Think Differently

```
You are a helpful assistant. Your task is to conceive a complex question inspired from the Paragraph,
while ensuring it is completely different from the example provided below. Prohibit the use of
expressions, question types, and initial verbs that are identical to those in the Examples provided.
Avoid phrases such as 'Based on', 'Given the information provided', 'Using the data' or any similar
expressions that suggest references to the Paragraph. command
### Counterexample:
<Example1>: {seed1}
<Example2>: {seed2}
<Example3>: {seed3}
<Example4>: {seed4}
<Example5>: {seed5}

### Paragraph:
{text}

### Question:
```

D.6 Self-Instruct Prompting Templates for Data Generation

Self-Instruct relies on the following prompting template in order to elicit the generation from language models.

```
Come up with a series of tasks:

Task 1: {instruction for existing task 1}
Task 2: {instruction for existing task 2}
Task 3: {instruction for existing task 3}
Task 4: {instruction for existing task 4}
Task 5: {instruction for existing task 5}
Task 6: {instruction for existing task 6}
Task 7: {instruction for existing task 7}
Task 8: {instruction for existing task 8}
Task 9:
```

Table 14: Prompt used for *Self-Instruct*

D.7 Faithfulness Evaluation

Table 15 shows the prompt to select more faithful instruction. The prompt originates from (Li et al., 2024) with minor modifications.

Table 15: Prompt for Faithfulness Evaluation (Li et al., 2024)

```
Below is an instruction from an user and a candidate answer. Let's think step by step. Evaluate
whether or not the answer is a good example of how AI Assistant should respond to the user's
instruction. Please assign a score using the following 5-point scale: 1: It means the answer is
incomplete, vague, off-topic, or not exactly what the user asked for. For example, some content
seems missing. Or the response is from another person's perspective with their personal experience
(e.g. taken from blog posts). Or it contains promotional text or other irrelevant information. 2:
(between 1 and 3) 3: It means the answer is helpful but not written by an AI Assistant. It addresses
all the basic asks from the user. It is complete and self contained with the drawback that the
response is not written from an AI assistant's perspective, but from other people's perspective.
For example, it contains personal experience or opinion, mentions comments section, or share on
social media, etc. 4: (between 3 and 5) 5: It means it is a perfect answer from an AI Assistant.
It has a clear focus on being a helpful AI Assistant, where the response looks like intentionally
written to address the user's question or instruction without any irrelevant sentences. The answer
provides high quality content, demonstrating expert knowledge in the area, is very well written,
logical, easy-to-follow, engaging and insightful.
Your reply should be only 1 or 2 or 3 or 4 or 5, without providing any reasoning and explanation.
### Instruction: {instruction}
### Answer: {response}
### Your Reply:
```

E Data Analysis

E.1 Quality, Length and Diversity

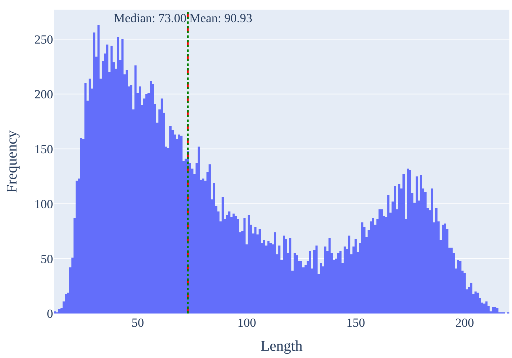

Figure 5: FANNO Instruction Length Distribution

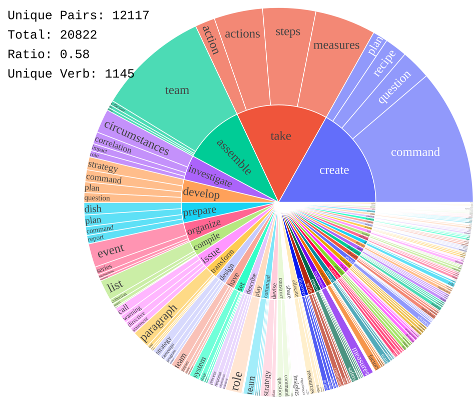

Figure 6: Top 50 common verbs and their corresponding nouns in FANNO

Figures 5 and 7 show the instruction length distribution of FANNO and Alpaca-Cleaned, respectively. It is worth noting that the mentioned length includes both the instruction and input combined. Figures 6 and 8 show the verb-noun diversity of FANNO and Alpaca-Cleaned, respectively. Figure 9 shows the comparison of quality and complexity between FANNO and Alpaca-Cleaned.

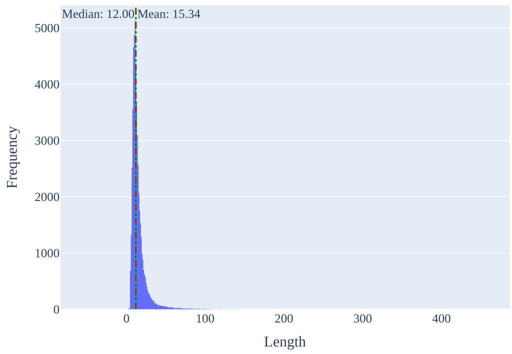

Figure 7: Alpaca-Cleaned Instruction Length Distribution

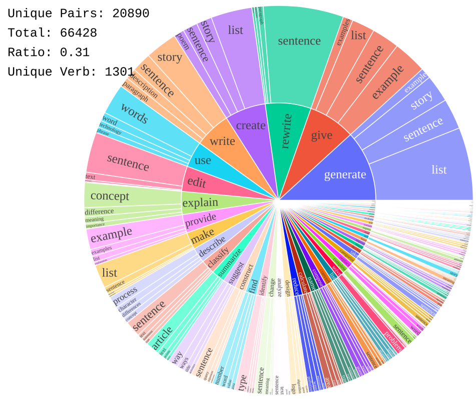

Figure 8: Top 50 common verbs and their corresponding nouns in Alpaca-Cleaned

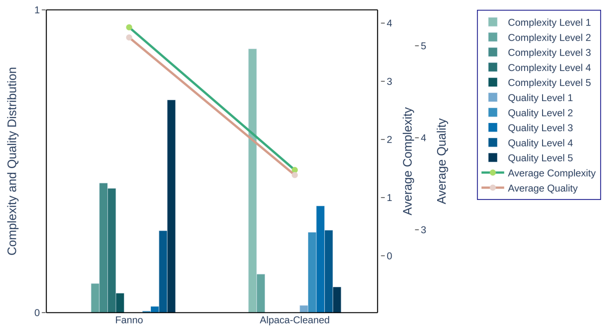

Figure 9: Quality and Complexity Comparison between FANNO and Alpaca-Cleaned

F Human Evaluation

F.1 Complexity Level

The first tier (0 point) pertains to instructions that exhibit apparent issues, such as being unanswerable or containing missing information. The second tier (1 point) involves instructions that can be answered using everyday knowledge. These instructions may assess basic skills, analyze human emotional experiences, or organize activities without requiring much specialized knowledge in a particular field. The third tier (2 points) comprises expert instructions. These necessitate specialized knowledge and require thorough deliberation steps to fulfill the instruction's requirement.

F.2 Instruction Complexity Human Evaluation

2 point (Expert Level)

1. Create a series of interactive exercises for a group of advanced French learners to practice the conditional tense, incorporating a variety of verb forms and sentence structures, while also encouraging them to engage in peer-to-peer learning and problem-solving. Consider using a combination of written and oral activities, and provide clear instructions and examples for each exercise. Additionally, design a system for assessing their progress and providing personalized feedback.

2. How can we optimize the WordPress website's performance for logged-in users without employing the Auto-Cache Engine? Consider various caching strategies and evaluate their potential impact on user experience and website functionality.

3. Design a multifaceted approach to streamline the patient registration process for a healthcare facility, ensuring adherence to ICD-10 and CPT coding standards, while providing exceptional customer

service to a diverse patient population. Consider implementing innovative technologies and collaborating with various departments to optimize workflows and enhance overall efficiency. Evaluate the potential impact of this approach on patient satisfaction, staff morale, and financial performance.

4. Assemble a team of data experts to evaluate the potential impact of a centralized data strategy on the decision-making process of a tech startup, considering the long-term benefits and potential drawbacks. Analyze various case studies of successful companies, such as Google, Apple, Amazon, and Facebook, to identify key strategies and best practices for implementing a data-driven culture. Evaluate the role of immediate returns versus long-term benefits in the adoption of data-driven decision-making and provide recommendations for managing potential challenges, such as data security and privacy concerns.

5. Assemble a team of nutritionists and chefs to devise a creative and nutritious menu for a charity gala, utilizing natural sweeteners as the primary ingredient in each dish, while ensuring that the final creations are visually appealing and can be prepared in large quantities. Additionally, consider the dietary restrictions of various attendees and incorporate alternative options for those with gluten, dairy, and nut allergies. The team should also aim to minimize food waste and maximize the use of locally sourced ingredients.

1 point (Everyday Level)

1. Develop a weekly routine that integrates both your professional and personal commitments, ensuring that you effectively manage your time and accomplish your goals. What unique strategies could you employ to optimize your productivity during your weekly review and planning session?

2. Persuade your employer to grant you the flexibility to work from home for a specified number of days per week, demonstrating the potential time and cost savings, as well as the potential benefits to your overall well-being.

3. Capture the essence of a cherished memory by taking a photograph of a cherished photograph. Ensure the image is visually appealing and evokes a sense of nostalgia.

4. Translate the following paragraph from English to another language of your choice. Ensure that the translation conveys the original meaning and intent. "Analyze the artworks displayed at the exhibition from various perspectives. Which artwork resonates the most with the theme of environmental conservation? Provide reasons for your answer."

0 point (Bad)

1. Utilize the data from the Neighbourhood Forum Launch event to determine the percentage of attendees who were members prior to the event and the percentage who joined during the event. Additionally, identify the top three focus groups with the highest number of attendees and determine the average number of attendees per focus group. Finally, calculate the total number of attendees who placed a dot on the Forum map and the percentage of attendees who did so.