Instruction Mining: High-Quality Instruction Data Selection for Large Language Models

Yihan Cao*
Carnegie Mellon University
Pittsburgh, PA 15213
yihanc@cs.cmu.edu

Yanbin Kang*
kybconnor@gmail.com

Lichao Sun†
Lehigh University
Bethlehem, PA 18015
lis221@lehigh.edu

Abstract

Large language models typically undergo two training stages, pretraining and finetuning. Despite that large-scale pretraining endows the model with strong capabilities to generate natural language responses, these pretrained models can still fail to understand human instructions at times. To enhance language models' ability of interpreting and responding to instructions, instruction finetuning has emerged as a critical method in this area. Recent studies found that large language models can be finetuned to perform well even with a small amount of high-quality instruction-following data. However, the selection of high-quality datasets for finetuning language models still lacks clear guidelines to follow. In this paper, we propose INSTRUCTMINING, a linear rule for evaluating instruction-following data quality. We formulate INSTRUCTMINING using specific natural language indicators. To investigate the relationship between data quality and these indicators, we further conduct extensive finetuning experiments. The experiment results are then applied to estimating parameters in INSTRUCTMINING. To further investigate its performance, we use INSTRUCTMINING to select high-quality data from unseen datasets. Results demonstrate that INSTRUCTMINING can help select relatively high-quality samples from various instruction-following datasets. Compared to models finetuned on unfiltered datasets, models finetuned on INSTRUCTMINING selected datasets perform better on 42.5% cases.

1 Introduction

As the cutting edge of natural language processing advances, large language models (LLMs) have demonstrated transformative capabilities, powering numerous applications with the strong ability in automatically generating responses according to human instructions. Nevertheless, it is hard sometimes for language models to capture the meaning of human instructions and respond to them even if they are pretrained with large amount of data. To counter this challenge, instruction tuning emerged as a paramount method in tailoring the behaviours of LLMs (Wei et al., 2021; Ouyang et al., 2022; Chung et al., 2022; Wang et al., 2022a). Instruction tuning leverages instruction-response pairwise data (henceforce referred to as instruction data) during finetuning. It facilitates the alignment of models with human preferences and their knowledge base, enabling the generation of desired outputs in response to various instructions. However, obtaining a large corpus of diverse, human-crafted instructions could be very expensive. To solve this problem, Wang et al. (2022a) proposed a methodology that prompts the model to generate its own instruction-following data for finetuning. Similarly, Taori et al. (2023) proposes Alpaca that employs GPT-3.5 (Ouyang et al., 2022) to create these instruction-following examples. Even though these methods could scale up the size of instruction-following data, they still inevitably consume a lot of resources.

[1]Equal contributions.

Preprint. Work in progress.

This problem makes researchers begin to explore whether using small quantity of high-quality instruction-following data can yield robust performance. Encouragingly, recent studies confirm that this approach holds promising potential. Zhou et al. (2023) proposes LIMA, which is instruction finetuned with human selected high-quality data. This study demonstrates its robust improvement over other language models of comparable size that are finetuned with unfiltered data. However, LIMA still requires human or machine experts to help set up a rule for selecting data from a large amount of data, which is time-consuming and expensive.

In this paper, we propose INSTRUCTMINING, a linear rule for selecting high-quality instruction data, which does not require human or machine annotation. We first propose our quality evaluation hypothesis that the quality of instruction data can be estimated using the loss generated by the finetuned model on a fair evaluation set, which contains unbiased human written instructions and high-quality responses. Under this hypothesis, we are able to quantify instruction dataset's quality using inference loss. However, estimating the inference loss requires us to actually finetune a language model, which could be time-consuming. To overcome this obstacle, we introduce a set of selected natural language indicators, which can be leveraged to predict the inference loss without actually finetuning an LLM.

To investigate the indicators' relationship with instruction data quality, we first sample 78 distinct subdatasets from a diverse data pool. Subsequently, we record each finetuned models' inference loss on the evaluation set. We then compute the indicator values across each subdataset. Finally we apply a statistical regression model to determine the relationship between the inference loss and indicator values, based on the rich experimental data we have obtained. We further demonstrate that INSTRUCTMINING is valid and scalable by comparing the inference performance between models finetuned using INSTRUCTMINING and random sampled datasets.

Our contributions are summarized as follows:

- In this paper, we are the first to provide a simple and explainable recipe for quantifying and selecting high-quality instruction-following data.
- We propose INSTRUCTMINING, a linear quality rule and bag of indicators for evaluating instruction-following data quality, and estimate the parameters in INSTRUCTMINING through extensive finetuning experiments on LLAMA-7B models.
- Comprehensive results show that INSTRUCTMINING can significantly improve finetuning performance. The model fine-tuned on filtered data performs better in 42.5% of the cases.

2 Methodology

2.1 What is Instruction Quality?

In this paper, we follow the superficial alignment hypothesis proposed by Zhou et al. (2023) that a model's knowledge is mostly learnt during pretraining, while instruction-following data teaches the model to follow a certain pattern when interacting with users. Hence, the quality of these instruction-following data could be viewed as its ability to efficiently steer language models in learning to generate responses in a particular manner. Based on this assumption, we further propose our instruction quality evaluation hypothesis as follows,

> **Instruction Quality Evaluation Hypothesis**: Given an instruction dataset D, we finetune a language model on D, denoted as $\tilde{M}$. The instruction quality of D can be estimated through the inference loss of $\tilde{M}$ on a evaluation dataset D_{eval}.

To ensure the inference loss provides a valid measure for evaluating data quality, the evaluation set should comprise a selected collection of unbiased and high-quality instruction-following samples.

In particular, given an instruction-following dataset D, we finetune a base language model M using D with model training settings S. S normally refers to training batch size, epochs, etc. The obtained finetuned language model is denoted as $\tilde{M}$. We define the dataset D's quality $Q_{D|M,S}$ as below,

$$Q_{D|M,S} \propto -L(\tilde{M}, D_{eval}) \tag{1}$$

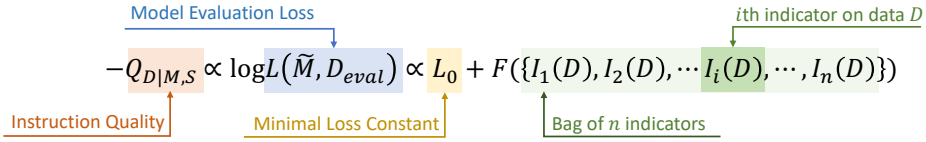
where D_{eval} refers to the high-quality and unbiased evaluation set, and $\propto$ means a direct proportion.

Indicator	Notation	Explanation
Length	Len	The average length of every response in the dataset.
Reward score	Rew	The average reward model inference score of every pair in the dataset. (Köpf et al., 2023)
Perplexity	PPL	The exponentiated average negative log-likelihood of response.
MTLD	$MTLD$	Measure of Textual Lexical Diversity McCarthy and Jarvis (2010)
KNN-i	KNN_i	Distance to approximate i^{th}-nearest neighbors (Dong et al., 2011) in SentenceBERT(Reimers and Gurevych, 2019) embedding space.
Unieval-naturalness	Nat	The score of whether a response is like something a person would naturally say, provided by the UniEval (Zhong et al., 2022) dialogue model.
Unieval-coherence	Coh	The score of whether this response serves as a valid continuation of the previous conversation, provided by the UniEval (Zhong et al., 2022) dialogue model.
Unieval-understandability	Und	The score of whether the response is understandable, provided by the UniEval (Zhong et al., 2022) dialogue model.

Table 1: Natural language indicators for instruction quality evaluation. Every data example is viewed as a pair of instruction(input) and response(output). Unless otherwise specified, the indicator value on a dataset is the average value of each sample on the indicator.

2.2 Quality Evaluation

According to Equation 1, we utilize the inference loss to evaluate instruction quality. However, finetuning an LLM for evaluation can be inefficient. To solve this problem, we introduce a set of natural language indicators and use the indicators to predict the inference loss. In this paper, We have a set of indicators $I = \{I_i, i \in N^*\}$[1], which is detailed in Table 1. For every given instruction-following dataset D, we compute the corresponding indicator values $I(D) = \{I_i(D), i \in N^*\}$. There exists a function F such that the aforementioned model inference loss $L(\tilde{M}, D_{eval})$ can be approximated using $F(I(D))$. The relationship between the finetuned model inference loss L and these computed indicators can be formulated as in Equation 2.



(2)

In this paper, we assume that there exists a multivariate linear function of $I_i, i \in \{1, \cdots, n\}$ that is proportional to the logarithmic loss. Consequently, Equation 2 can be reparameterized as Equation 3:

$$\begin{aligned} \log L(\tilde{M}, D_{eval}) &\propto L_0 + F\{I(D)\} \\ &\propto L_0 + \beta_0 + \beta_1 I_1(D) + \beta_2 I_2(D) + \cdots + \beta_n I_n(D) + \epsilon \end{aligned} \tag{3}$$

where β_0 signifies the linear constant and $\beta_i, i \in N^*$ represent a sequence of linear coefficients. ϵ refers to the random error term.

To investigate the relationship between these indicators and the overall dataset quality, it becomes necessary to accumulate experimental results to estimate the unknown parameters $\beta_i, i \in N$. In this study, we employ the Least Squares method (Björck, 1990) to estimate the parameters in the multivariate function. The Least Squares method is a standard approach in regression analysis for the approximate solution of overdetermined systems. The technique minimizes the sum of the square residuals, thus providing the optimal fit between the observed and predicted data in terms of reducing the overall prediction error. Our experiment result and analysis are detailed in Section 4.

[1] N^* refers to natural numbers starting from 1, and N refers to natural numbers starting from 0.

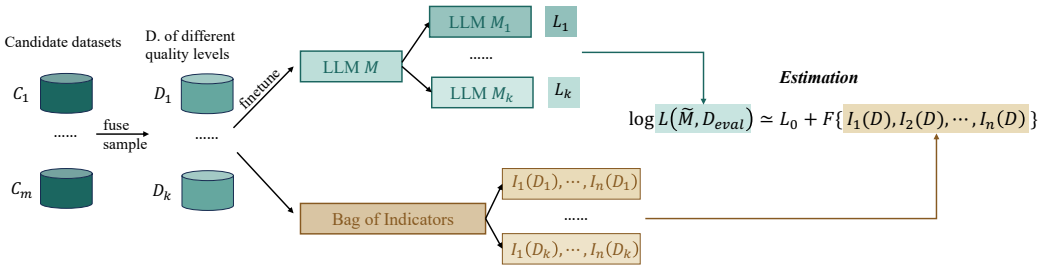


Figure 1: Our empirical study procedure. We first select several candidate datasets. Then we fuse and sample from them to form datasets of different quality levels. For each dataset, we finetune a language model on it and evaluate the model on a shared evaluation set. We also calculate bag of indicator values on the dataset. Finally, we perform a linear regression analysis based on our curated experiment results to estimate the linear rule parameters.

2.3 Empirical Study Design

In this paper, we design experiments to investigate both multivariate and univariate correlations between indicators and dataset quality. The key distinction between the two exists in the finetune data sampling strategy. For multivariate evaluation experiments, we randomly sample subdatasets from several candidate datasets of different preassumed quality levels. Conversely, for univariate experiments, we sample fine-grained subdatasets according to their indicator values.

2.3.1 Multivariate Evaluation

The general procudure of our multivariate evaluation experiment is shown in Figure 1. To estimate the correlation between evaluation loss L and bag of indicators I and promise the scalability of our method, we need to get datasets of different indicator values. To achieve this, we commence by selecting several commonly used datasets with different presumed quality levels and fuse them together with randomly sampled percentages to create finetune datasets. These sampled finetune datasets should encompass varying proportions of preassumed high quality and low quality examples. For each of these sampled datasets D_i, we compute its respective indicator values $I(D_i)$ and finetune a base language model M using D_i. Following Equation 1, the quality Q_{D_i} for dataset D_i is approximated using the evaluation loss of finetuned model $\tilde{M}_i$ on a fair evaluation dataset D_{eval}. Following the collection of a range of results correlating Q_{D_i} with $I(D_i)$, we undertake a statistical regression analysis to discern patterns and relationships within the compiled data.

2.3.2 Univariate Evaluation

In addition to multivariate evaluation, we also study the individual correlation between each indicator and instruction data quality. Except for the randomly selected datasets from multivariate experiments, we also conduct fine-grained sampling here, wherein subdatasets are sampled based on their specific indicator values. For a given indicator I_j, we sample a series of subdatasets of the same size with fine-grained indicator values. To illustrate, consider indicator Rew. We compute the score for each sample in the dataset and then rank them from lowest to highest. Subsequently, these data samples are separated into K tiers, according to their respective Rew scores. For each dataset, we finetune a base model and conduct the evaluation to obtain a quality estimation of the sampled instructiond dataset. The correlation analysis tools are then employed to evaluate the quality value of each dataset in relation to the indicator value.

3 Empirical Settings

3.1 Datasets

Candidate Training Datasets. In order to create diverse training datasets, we collect data from various sources with differing collection standards. This approach ensures that the datasets exhibit theoretical differences in quality and maintain diversity among sources. For this purpose, we have

selected the following datasets as candidate datasets: ALPACA (Taori et al., 2023), OPEN-ASSISTANT (Köpf et al., 2023), STACKEXCHANGE, and WIKIHOW. Due to the varying formats, sizes, and distributions of different datasets, we have applied distinct processing procedures to each dataset. For detailed processing procedures, please refer to Appendix A. Table 2 provides an overview of the candidate training datasets after preprocessing.

Datasets	Sourced from	Size	Quality
ALPACA	Generated w/ davinci	52.0k	Normal
OPEN-ASSITANT	human-generated	3.4k	Both
STACKEXCHANGE	human-generated	3.0k	High
WIKIHOW	human-generated	2.0k	High

Table 2: Overview of the candidate training datasets after preprocessing.

Evaluation Datasets. To address real-world requirements, we diversified the instructions by combining test data from different evaluation datasets, which includes 252 instructions from Wang et al. (2022a) and 80 from Zheng et al. (2023).

In our study, we employed `gpt-3.5-turbo` from OPENAI to generate five unique outputs for each instruction. We chose `gpt-3.5-turbo` due to its superior performance in instruction-following compared to LLAMA-7B. To account for the possibility of multiple valid outputs for a given instruction, we generated multiple outputs to represent better the output distribution of `gpt-3.5-turbo`.

Sampling Candidate Datasets. As mentioned in section 2.3, we leverage two different fusing method for **multivariate** and **univariate** analysis. We merged candidate training datasets, resulting in each dataset containing 2,000 instruction-output pairs.

Sampling strategy for multivariate analysis. We generated a random number r_i for each dataset and randomly selecting $2000 * r_i / \sum_i r_i$ samples from each dataset for combination.

Sampling strategy for univariate analysis. We merged all the data and sorted it according to the target indicator. We then evenly selected K quantiles as starting points on sorted data and extracted 2000 following consecutive data to form a new dataset. This selection process ensures diverse results of the target indicator on these datasets. We used K=8 in our experiments.

Considering the significant size difference between the ALPACA and other datasets used, we randomly sampled 2000 data from ALPACA to maintain scale consistency across all the candidate datasets.

3.2 Finetuning Settings

We conduct all instruction tuning on the same base model LLAMA-7B (Touvron et al., 2023). All finetuning datasets are of the same size, 2000 examples in each. We apply 8bit QLORA (Dettmers et al., 2023) on W_q, W_k, W_v in the attention module with LoRA (Hu et al., 2021) $r = 8, \alpha = 32$. We run model finetuning for 3 epochs, with per step batch size set to 8. We use Adam with $\beta_1 = 0.9, \beta_2 = 0.999$, and linear learning rate scheduler starts from $5e-5$, decays to 0.

Each finetuned model is evaluated on the evaluation dataset mentioned in section 3.1. We run all finetuning and evaluation experiments on 6 NVIDIA RTX A6000.

4 Empirical Results

In this section, we detail our core findings with experiment results. Following section 2.3, we analysis the relationship between indicators and data quality from two perspectives, multivariate and univariate. Section 4.1 presents our experimentation results and analysis on randomly sampled subdatasets. Section 4.2 elaborates on our analysis on the correlation between single indicator and instruction quality, and section 4.3 shows how to use our resulted quality evaluation function to select high-quality data, and compare our method with baselines.

Variable	Coef.	Std err.	t value	$P > \|t\|$
β_0	0.1840	0.565	0.325	0.746
β_{PPL}	0.0236	0.022	1.080	0.283
β_{MTLD}	0.0003	0.004	0.086	0.932
β_{Rew} **	-0.0828	0.035	-2.361	0.020
β_{Len} *	0.0001	7e-5	1.907	0.060
β_{Nat}	0.9149	1.442	0.635	0.527
β_{Coh}	0.3784	0.647	0.585	0.560
β_{Und}	-0.3409	1.618	-0.211	0.834
β_{Knn_6} ***	-1.0611	0.340	-3.124	0.002

[1] R^2=0.848, Adjusted R^2=0.835, F-statistic=64.99.
[2] Prob(F-statistic)=9.75e-35, Log-Likelihood=272.77.
[3] *: $p \leq$0.1, **:$p \leq$0.05, ***:$p \leq$0.01.

(a) OLS regression results including all variables.

Variable	Coef.	Std err.	t value	$P > \|t\|$
β_0 ***	1.0694	0.070	15.338	0.000
β_{PPL}	-	-	-	-
β_{MTLD}	-	-	-	-
β_{Rew} ***	-0.1498	0.007	-21.141	0.000
β_{Len} ***	8e-5	8e-6	9.759	0.000
β_{Nat}	-	-	-	-
β_{Coh}	-	-	-	-
β_{Und}	-	-	-	-
β_{Knn_6} ***	-0.9350	0.072	-12.992	0.000

[1] R^2=0.838, Adjusted R^2=0.833, F-statistic=169.3.
[2] Prob(F-statistic)=1.24e-38, Log-Likelihood=268.52.
[3] *: $p \leq$0.1, **:$p \leq$0.05, ***:$p \leq$0.01.

(b) Stepwise OLS regression results.

Table 3: Linear regression parameter estimation results using ordinary least squares (OLS). $P > |t|$ represents p value under student test on each coeficient. Lower p value indicating that the coeficient for this variable is more significant and acceptable. R^2 and adjusted R^2 represents how well the data is fit using the estimated linear function.

4.1 Multivariate Analysis

We randomly sampled 78 datasets from candidate dataset with different percentages.Each comprises varying proportions of high-quality and low-quality examples. The experiment results is shown in Appendix B. We first analyze the distribution of each indicator to make sure that our data satisfies multivariate linear regression assumptions. The distribution figures for each indicator is shown in Appendix B. We also run Kolmogorov-Smirnov test (Massey Jr, 1951) on each indicator and collected loss values to make sure that all variables follow normal distribution.

We use stepwise regression to perform the analysis. Stepwise regression is a step-by-step iterative method applied to multivariate linear regression, to help find significant variables. We run the regression on variables mentioned in Table 1. The regression results before and after stepwise regression are available in Table 3.

As shown in Table 3, we delineate our estimated evaluation function, which is articulated as Equation 4. According to the analysis result, reward score and nearest neighbour score, which is a metric of dataset diversity, are the most significant indicators to the general instruction data quality.

$$\begin{aligned} &Q_{D|M,S} \propto -L(\tilde{M}, D_{eval}) \\ &\log L(\tilde{M}, D_{eval}) \propto 1.0694 - 0.1498 Rew + 8.257 * 10^{-5} Len - 0.9350 Knn_6 + \epsilon \end{aligned} \tag{4}$$

4.2 Univariate Analysis

For every indicator, we analyze its linear correlation with evaluation loss. The univariate analysis results are shown in Figure 2. The two series demonstrate similar patterns. Variables PPL, $MTLD$, Nat, and Und exhibit positive correlations with the anticipated evaluation loss, suggesting an increase in these variables may result in a higher evaluation loss. Conversely, Rew and Coh showcase negative correlations with the evaluation loss, implying that an increase in these variables might lead to a reduction in loss. Comparing to randomly selecting examples from the data pool, it is obvious that selecting datasets directly according to PPL, $MTLD$, Rew are more preferable. Notably, this does not conflict with our analysis in multivariate analysis, since there are multiple indicators which are prominent to the fluctuation of inference loss and these indicators can share multicollinearity.

4.3 Quality-Guided Instruction Selection

In this section, we follow the regression result in Equation 4 to select high quality examples from an unseen dataset, `databricks-dolly-15k` [2].

[2]https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

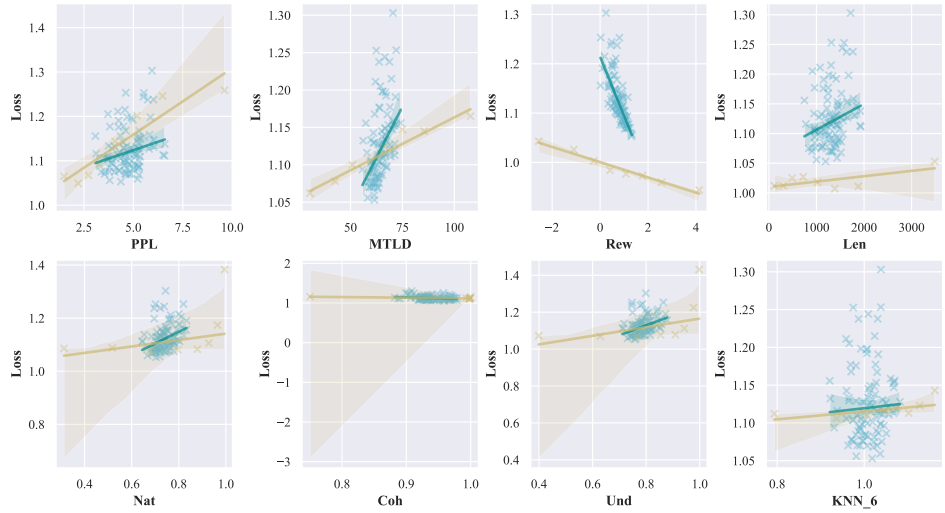


Figure 2: Univariate analysis regression plot. For every indicator, we plot its indicator value w.r.t. the actual inference loss. Series cyan represents data collected from multivariate analysis (randomly sampled) while series yellow represents data collected from univariate analysis (hierarchically sampled). For every cluster we estimate a univariate linear function between loss and indicator. The regression confidence level is 95%.

Sampling Method	**Notation**	$\exp(Rule)$	$Rule$	**Loss(epoch 1)**	**Loss(epoch 2)**	**Loss(epoch 3)**
Fine-grained	E_1	1.026	0.0260	1.383	1.375	1.371
	E_2	0.975	-0.025	1.377	1.368	1.370
	E_3	0.850	-0.163	1.366	1.364	1.359
	E_4	**0.749**	**-0.289**	**1.362**	**1.352**	**1.349**
Random	E_5	1.205	0.187	1.384	1.372	1.367
	E_6	1.193	0.177	1.368	1.363	1.357

Table 4: Quality-guided instruction selection experiment result. $Rule$ refers to the expected evaluation loss estimated using our quality rule. We train three epochs on each dataset, and record the actual loss value after each epoch.

For selected datasets, we finetune the base model LLAMA-7B and compare our finetuned model to baseline models. Experiment results show that our formulation of instruction data quality is valid and is scalable to other instruction-following datasets. We present our experiment results in Table 4 and Figure 3. To be noticed, Table 4 shows the results of multiple models finetuned on selected datasets, and Figure 3 presents the GPT evaluated comparison result between the selected high-quality dataset and randomly sampled dataset. All evaluation experiments are done on the evaluation set mentioned in section 3.1.

From Table 4, we can see that with higher expected evaluation loss, the actual loss is also higher, which indicates that the instruction data is be of lower quality. Notably, the loss gap among E_2, E_3 and E_4 is comparatively larger than the gap among E_1,

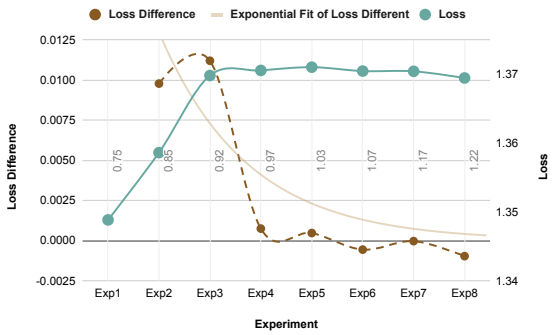


Figure 3: Actual evaluation loss and loss difference w.r.t. experiments. The grey number in the figure refers to the $\exp(Rule)$ value for each experiment. The experiments are ranked in descending order according to their estimated quality values. The left axis refers to loss different and the right axis refers to actual loss values.

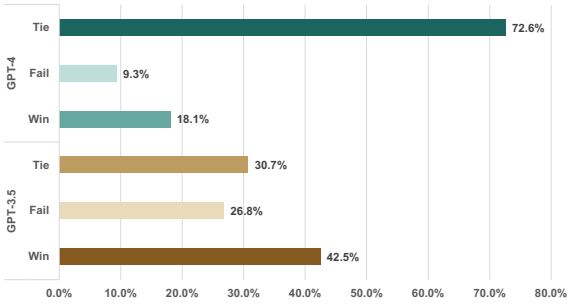


Figure 4: GPT evaluation comparison between rule selected dataset and randomly sampled dataset. In the context of this figure, *Win*, *Fail* and *Tie* here denote comparative outcomes when the generation results of selected dataset finetuned model are evaluated against those of a randomly sampled dataset finetuned model.

E_2 and E_3. We specifically study the pattern of the loss difference among the fine-grained datasets. As shown in Figure 3, change of loss difference between the datasets of high quality is significantly larger than which between datasets of low quality. This reveals that the performance of instruction finetuning on LLAMA-7B models might be very sensitive of very high quality data.

Except for analysis based on fine-grained datasets, we also evaluate our method by comparing the finetuning performance between the seleted data and randomly sampled data. Results are shown in Table 4 and Figure 3. Table 4 evaluates from the perspective of inference loss while Figure 3 leverages `gpt-3.5-turbo` and `gpt-4` to compare the answer generated by the two finetuned models. The results show that our selected datasets perform better than the randomly sampled dataset. However, the difference between the two is not very large according to `gpt-4`. This may be attributable to the relative diminutiveness of our base model, implying that its foundational knowledge may be insufficient for addressing relatively complex instructions. Except for this, our method leverages `gpt-3.5-turbo` to generate responses for the golden evaluation set, which make our method more aligned with `gpt-3.5-turbo`.

5 Related Work

Recent studies have proposed instruction tuning methods for fine-tuning Large Language Models (LLMs), demonstrating their generalization capabilities for unseen instructions Wei et al. (2021). To enhance instruction tuning, some researchers have focused on increasing the data size through various methods Honovich et al. (2022)Wang et al. (2022a). In contrast, others have shown that a smaller amount of high-quality instruction data can yield effective models Zhou et al. (2023). Reinforcement learning from human feedback (RLHF) has been used to align language models with human intent Ouyang et al. (2022). Our method focuses on limited data and leverages the reward model from RLHF to estimate instruction quality.

The field has seen growth with the publication of numerous instruction datasets(Taori et al. (2023), Köpf et al. (2023), Honovich et al. (2022)). Several works(Chung et al. (2022)) have combined multiple datasets to increase the amount and diversity of instruction data, resulting in performance gains. Evidence from Iyer et al. (2023), Wang et al. (2023), Wang et al. (2022b), and Longpre et al. (2023) suggests that enhancing instruction diversity can significantly improve instruction tuning performance. A quality gap exists between different data sources due to varying data collection methods across instruction datasets. Some studies, such as Gunasekar et al. (2023), have demonstrated that increasing the proportion of high-quality data can enhance performance.

Lee et al. (2023) shows that the recently proposed Task2Vec Achille et al. (2019) diversity is a reliable diversity coefficient for LLMs' dataset, an aspect of quality assessment. Our work aims to present an instruction quality evaluation method for measuring the quality of instruction datasets.

Other works have focused on estimating the quality of prompts, such as Gonen et al. (2022), which uses perplexity for prompt selection. Similarly, we test the perplexity with instruction quality.

6 Discussion and Future Work

In this paper, we propose a quality evaluation rule specifically for instruction finetuning. Extensive experiments have been conducted to estimate the parameter in this rule and to prove that our evaluation rule is valid and scalable to other datasets. However, limitations still exist in this work. First of all, in our experiments, we only include limited amount of simple indicators from previous works. Recently, many researchers are beginning to explore instruction diversity, which we could later include into our future work. Secondly, our method is only experimented on single-turn instruction-following and mostly human written datasets. We haven't tested on multi-turn and more complex conversational datasets. Finally, in this paper, we only study the relationship between indicator values and inference loss value on fixed base model, LLAMA-7B. As the next step of this work, we will expand our analysis to larger models, e.g. LLAMA-13B and LLAMA-65B. We will also include more evaluation sets and instruction datasets for further analysis.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning.

Åke Björck. 1990. Least squares methods. *Handbook of numerical analysis*, 1:465–652.

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*.

Databricks. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. Blog post.

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

Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586.

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

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

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

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.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2023. Opt-iml: Scaling language model instruction meta learning through the lens of generalization.

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. 2023. Openassistant conversations – democratizing large language model alignment.

Alycia Lee, Brando Miranda, and Sanmi Koyejo. 2023. Beyond scale: the diversity coefficient as a data quality metric demonstrates llms are pre-trained on formally diverse data.

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. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Frank J Massey Jr. 1951. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Philip M McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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.

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, 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. 2023. Llama: Open and efficient foundation language models.

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. 2023. How far can camels go? exploring the state of instruction tuning on open resources.

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, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation.

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

Appendix

A Instruction Datasets Details

Alpaca (Taori et al., 2023) is a dataset of 52,000 instructions generated by OpenAI's text-davinci-003 engine. We use the same prompt template as ALPACA.

Open Assistant (Köpf et al., 2023) is a project of a chat-based and open-source assistant. We use "timdettmers/openassistant-guanaco" dataset from HuggingFace (Wolf et al., 2020). We then filter out instances containing multiple turns to focus on single-turn dialogues and retain only English dialogues. The content following "### Human: " serves as the instruction, while the content after "### Assistant: " is considered the output.

Stack Exchange[1] is a network of question-and-answer websites. The voting mechanism on the site (upvote or downvote) is leveraged to maintain content quality. We started with "HuggingFaceH4/stack-exchange-preferences" on HuggingFace datasets. Referring to Zhou et al. (2023), We process the data considering both quality and diversity. We selected answers with the highest votes while filtering out those with less than or equal to 5 votes. We also removed HTML tags in the answers and excluded responses with character counts below 200 or above 4000. Applying these processes ensures the selected data's quality. To control diversity, only 20 answers per exchange from the total 179 on Stack Exchange were chosen based on meeting the quality criteria.

wikiHow [2] is a worldwide collaborative platform to teach people how to do anything. We employed a dataset sourced from Huggingface datasets. We utilized titles as prompts and the corresponding bodies as responses. To ensure dataset diversity, we extracted embeddings from the titles using the sentence transformer (Reimers and Gurevych, 2019), subsequently applying K-means clustering to group them into 19 categories. We then randomly select a category and select a data sample from that category, resulting in a dataset comprising 2000 instructions dataset.

Dolly (Databricks, 2023) is a dataset contain 15,000 human-generated insturction-following authored by Databricks employees. We use "databricks/databricks-dolly-15k" from Huggingface datasets and apply the prompt template from ALPACA

B Descriptive Analysis

We present our descriptive analysis on 78 randomly sampled results here.

Variable	Mean	Std.	Min	Median	Max
Loss	1.126	0.049	1.053	1.115	1.303
PPL	4.734	0.745	3.080	4.680	6.591
Knn_6	1.009	0.034	0.921	1.009	1.082
Len	1313.762	258.823	746.074	1309.738	1932.745
$MTLD$	64.406	3.891	55.752	64.047	74.190
Rew	0.776	0.285	0.017	0.785	1.328
Coh	0.939	0.022	0.882	0.935	0.979
Nat	0.738	0.039	0.645	0.736	0.833
Und	0.785	0.035	0.711	0.781	0.879

Table 5: Descriptive statistical analysis results on linear regression variables.

[1] https://stackexchange.com/

[2] https://www.wikihow.com/Main-Page

The distribution for every indicator is shwon in Figure 5. To make sure our OLS result is valid, we also perform Kolmogorov–Smirnov test to see whether the variables follow normal distribution. Results show that all variables in our analysis follow normal distribution, which satisfies the OLS regression assumption.

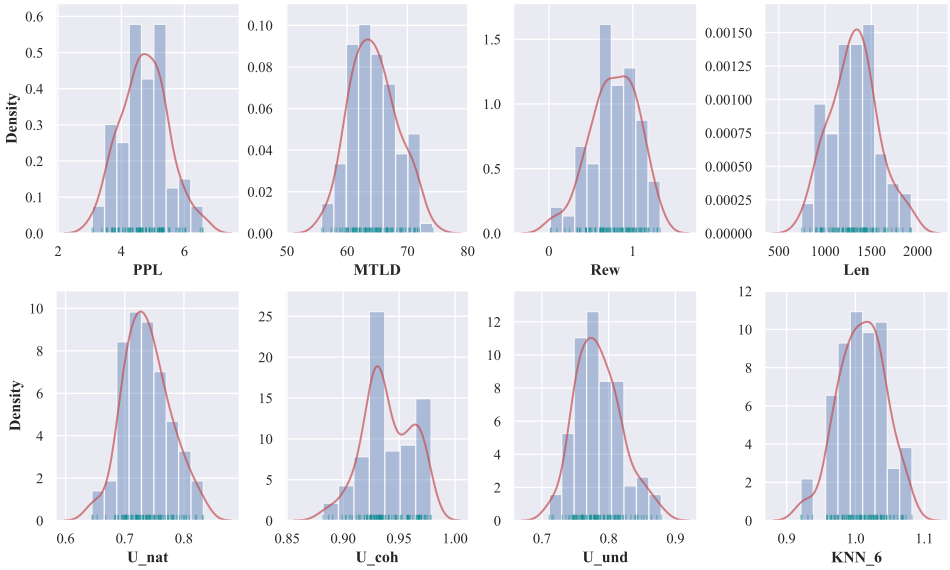


Figure 5: Distribution for all indicators.

Variable	Statistic	p Value
PPL	0.999	0.000
$MTLD$	1.000	0.000
Rew	0.583	0.000
Len	1.000	0.000
Nat	0.740	0.000
Coh	0.811	0.000
Und	0.761	0.000
KNN_6	0.821	0.000
Loss	0.854	0.000

Table 6: KS test results for all variables in linear regression. Smaller p value indicates that the variable is highly possible to follow normal distribution.