Bread: A Hybrid Approach for Instruction Data Mining through Balanced Retrieval and Dynamic Data Sampling

Xinlin Zhuang[1], Xin Mao[2], Yuan-Hao Jiang[1], Hongyi Wu[1], Shangqing Zhao[1], Li Cai[1,3], Shu Liu[1], Yang Chen[1], Yuxiang Song[1], Chenghao Jia[1], Yuhao Zhou[1], and Man Lan[1⋆]

[1] School of Computer Science and Technology, East China Normal University, China
[2] Nanyang Technological University, Singapore
[3] Guizhou University, China
xinlinzhuang@stu.ecnu.edu.cn mlan@cs.ecnu.edu.cn

Abstract. Recent advancements in Instruction Tuning (IT) have shown promise for aligning Large Language Models (LLMs) with users' intentions, yet its efficacy is often compromised by dependence on high-quality datasets. Previous works have concentrated on the aggregation or production of huge IT datasets through human labor or significant cost-intensive LLM APIs, which lacks adequate mechanisms to guarantee the quality of the resulting data. Moreover, training on such amount of IT data is both time-consuming and costly. To address these issues, we present **Bread** (Instruction Mining through **B**alanced **RE**trieval **A**nd **D**ynamic Data Sampling), a novel approach designed to minimize the requisite volume of IT data. Bread uses a two-stage strategy combining balanced retrieval and dynamic sampling to focus on data diversity and quality, offering a cost-saving solution without relying on any specific LLMs. Experimental results suggest that Bread outperforms baselines and shows great flexibility across various IT datasets and LLMs, thereby marking a step forward in efficient Instruction Tuning. Our code is available at https://github.com/mihara-bot/Bread.

Keywords: Large Language Models · Instruction Tuning · Data Selection

1 Introduction

Instruction Tuning (IT) involves training with instruction data pairs to function as an efficacious technique for improving the proficiency of Large Language Models (LLMs) in following user directives and enhancing controllability, which effectively mitigates the discordance between the pre-training objectives of LLMs and the actual intentions of users, thereby unlocking the potential of LLMs across an extensive array of fields such as Law [25], Medicine [30], Finance [19], as well as a variety of NLP tasks [40]. The general pipeline of IT is shown in Fig. 1. To be specific, IT utilizes data in the *(Instruction, Output)* format to fine-tune

⋆ Corresponding author.

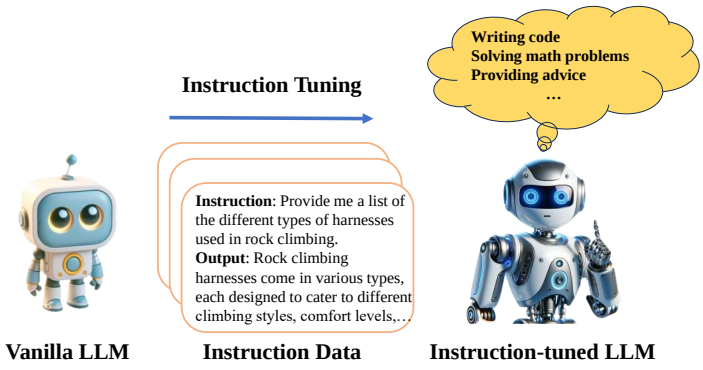

Fig. 1. Instruction Tuning involves training pre-trained vanilla LLMs on instruction datasets in supervised manner to enhance their capability to follow instructions.

LLMs [38]. Herein, *Instruction* is the specific guideline issued by a human to the model, while *Output* signifies the expected output.

Currently, IT is confronted with two significant challenges from the perspective of data. On the one hand, there is a notable lack of IT datasets with broad applicability, and most are highly specialized, suitable only for certain domains, which limits the ability of LLMs to generalize across various tasks. On the other hand, when LLMs are fine-tuned using low-quality data, there is an increased risk of the models generating inaccurate or imaginary content, a problem often referred to as *hallucination*, regardless of the size of the datasets used [15]. Additionally, recent works [41,14] have highlighted the importance of data **quality** rather than quantity in IT, especially evidencing substantial outcomes through merely **1,000** manually curated high-quality data examples [41].

To address the scarcity of IT datasets with broad applicability, researchers have implemented three primary strategies to cultivate more generalized datasets: collecting data via human contributors [31,7], generating data through LLMs [27,22,34,32,26], and a hybrid method that merges human data collection with LLM generation [16]. Despite these efforts, creating instruction datasets of sufficient scale, often ranging from tens to hundreds of thousands of examples, has demanded significant investments in terms of time and computational resources.

In an attempt to overcome the challenges associated with the abundance of low-quality IT data, researchers have proposed various methods aimed at downsizing the total amount of data required for IT training through selective filtering or sampling techniques [5,17,42]. These methods, while helpful, have certain drawbacks. Alpagasus [5] leverages advanced LLMs such as ChatGPT and Claude to assess the quality of instruction data samples. The reliance on these additional models means that Alpagasus may overlook the intrinsic capabilities of LLMs and also incurs high costs due to API calls for closed-source

models. Cherry [17] proposes a new metric known as Instruction Following Difficulty (IFD), which measures how challenging the instruction data samples are for specific LLMs, which involves extensive computation. Lastly, DQ [42] employs a general data sampling strategy that builds on GraphCut [12], concentrating on the data distribution aspect for keeping data quality. However, it falls short in enough control for ensuring data diversity.

To address these issues, we propose **Bread** (Instruction Mining through **B**alanced **RE**trieval **A**nd **D**ynamic Data Sampling), a two-pronged approach that promises no reliance on additional LLMs and reduces computational complexity. Bread consists of two core stages: **Diverse Data Retrieval** and **Dynamic Data Sampling**. The initial stage is marked by the employment of embedding-driven clustering to retrieve data, purposely filtering out superfluous instances to guarantee a diverse dataset, which is pivotal in striking a balance between the diversification of filtered data and minimization of redundancy. Subsequently, the second phase employs a dynamic data sampling method aimed at further condensing the dataset size. This technique prioritizes data representativeness while simultaneously sustains diversity, ensuring that the core characteristics of the dataset are not compromised. With an extensive range of applicability, Bread exhibits robust versatility and scalability across diverse LLMs, demonstrating flexibility in adapting to the characteristics of different models. Moreover, it can handle datasets from a multitude of sources (whether from human collection or LLM generation), revealing an inherent ability for seamless integration and usability. Our contributions are as follows. **Firstly**, we propose Bread, a novel instruction data mining approach based on balanced retrieval and dynamic data sampling. Bread outperforms baseline methods in almost all cases under different settings. **Secondly**, Bread can retain **10%** of the original large-scale IT datasets while preserving a substantial amount of their diversity and representativeness, thereby reducing costs and computational resources for training LLMs. **Finally**, Bread generalizes well to wide IT datasets (whether from collection or generation) and LLMs, thereby marking a step forward in efficient IT.

2 Related Work

2.1 Instruction Tuning

As a novel paradigm of Fine-tuning, Instruction Tuning enhances LLMs' performance by leveraging *(Instruction, Output)* pairs. Within this framework, the term *Instruction* specifies the human directive to the model, while *Output* encapsulates the model's expected response in accordance with the *Instruction*.

Currently, the tuning methods in IT remains somewhat static, dominated by approaches such as fine-tuning and Parameter-Efficient Fine-tuning, exemplified by techniques like LoRA [18]. Consequently, the formulation of IT datasets has garnered significant research focus. At present, three principal methodologies prevail for constructing instruction tuning data: Human Collection [31,7], LLM

Generation [27,22,34,32,26], and a hybrid of Human Collection and LLM Generation [16]. The Human Collection approach principally adapts conventional NLP tasks into a standardized format of instruction pairs. In contrast, LLM Generation involves devising various prompting mechanisms to harness LLMs' capabilities in generating appropriate data instances. Considering the problem of human errors and LLM hallucination [15,23,39,11] as well as the huge cost for tuning using huge amounts of data, prior works have attempted to reduce the instruction data size through a few strategies [5,17,42]. Compared to these strategies, Bread is not reliant on supplementary LLMs, exhibits minimal computational complexity, and effectively balances between data diversity and quality.

2.2 Data-centric AI

In the preceding decade, the domain of data-centric artificial intelligence (data-centric AI) has witnessed extraordinary developments. A series of seminal contributions have markedly propelled the field forward [6,21,36,37]. The foundational premise underpinning data-centric AI is the recognition that the caliber of data is of equivalent significance as algorithmic advances in the machine learning (ML) spectrum. For a finer point of clarification, it becomes vital that methodologies for data cleaning and mining showcase a heightened propensity for automation and flexibility [36,37]. The introduction of LLMs like ChatGPT heralds a new transformation, serving as a beacon for pivotal shifts across NLP applications. Presently, the substantial majority of LLMs have adopted the potent Transformer framework [29], which utilizes layers of transformer encoders or decoders to enhance data processing proficiencies. This transformative leap constitutes a distinguished milestone, underscoring the growing prominence of quality data in tandem with the architectural progression of models. Our work is directed toward the meticulous curation of high-quality datasets, favoring this approach over the mere accumulation of extensive data quantities.

3 Method

As shown in Fig. 2, Bread represents a two-stage data mining strategy merging Diverse Data Retrieval and Dynamic Data Selection. In the first stage (Section 3.1), embedding-based Clustering assisted with Perplexity (PPL) score ranking is adopted to gather valuable diverse data samples, with extraneous data being pruned. This is followed by iterative dynamic sampling (Section 3.2) to maintain dataset representativeness while ensuring variety for further selection.

3.1 Diverse Data Retrieval

Aligned with contemporary research findings [1,33], which highlight the enhancement of In-Context Learning capabilities in LLMs through diversity-based methods and the performance improvement brought by more diverse datasets in IT [41], our initial phase employs a semantic diversity-oriented strategy for data

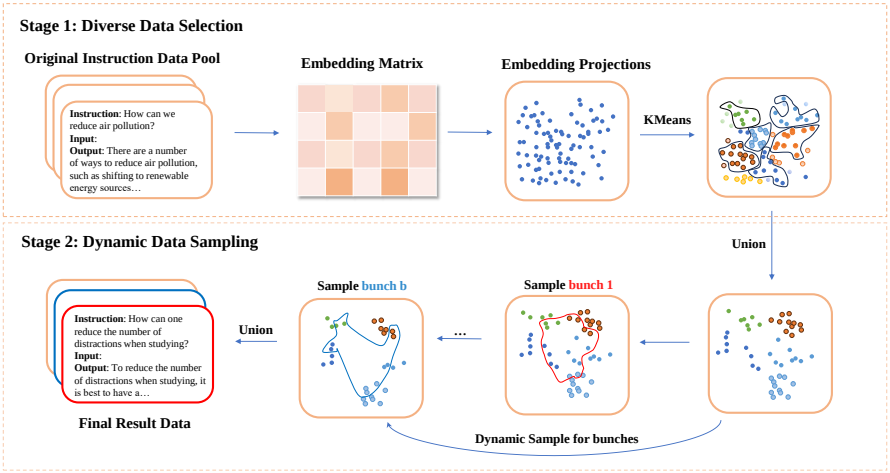

Fig. 2. The overview of Bread. Stage 1 involves assembling datasets characterized by high diversity, followed by iterative dynamic sampling to retain the most representative samples while preserving diversity within the dataset in Stage 2.

Algorithm 1 Algorithm for Diverse Data Retrieval

Input: List of instruction data embeddings E and correspoding PPLs P, number of samples per cluster n, number of clusters k, sampling thresholds th_{low} and th_{high}.
Output: Retrieved dataset with balanced diversity $\mathcal{D}'$.

Initialize an empty set $\mathcal{D}' = \{\ \}$
Apply KMeans Clustering to E to partition into k clusters
for $i = 1$ to k **do**
    Identify and collect embeddings E_i that belong to cluster i
    Sort samples within cluster i based on P
    Determine the middle confidence range using th_{low} and th_{high} percentiles
    Uniformly sample n embeddings from the middle confidence range
    Append the corresponding samples to $\mathcal{D}'$
end for
return $\mathcal{D}'$

Algorithm 2 Algorithm for Dynamic Data Sampling

Input: Instruction dataset obtained through Dynamic Data Sampling $\mathcal{D}'$, list of corresponding embeddings E, overall sampling ratio r, number of bunches b.
Output: Final sampled dataset $\mathcal{D}_f$.

Initialize index set $I = \{0, 1, \ldots, |\mathcal{D}'| - 1\}$, collection of bunches $\mathcal{B} = \{\}$
Compute final sample size $p = |\mathcal{D}'| \times r$
for $i = 1$ to b **do**
    Select subset $\mathcal{S}_i$ of size $\frac{|\mathcal{D}'|}{b}$ from $\mathcal{D}'$ via Eq. 2
    Update bunches $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{S}_i$
    Remove indices of samples in $\mathcal{S}_i$ from I and corresponding embeddings from E
end for
Initialize selected index set $\mathcal{S} = \{\ \}$
for subset $\mathcal{S}_i$ in $\mathcal{B}$ **do**
    Compute target size t_i for $\mathcal{S}_i$ via Eq. 3
    Uniformly sample t_i indices based on target count from $\mathcal{S}_i$ and append to $\mathcal{S}$
end for
return $\mathcal{D}_f = \{\mathcal{D}'[i] | i \in \mathcal{S}\}$.

retrieval. Utilizing KMeans Clustering [13,24] on embeddings of instrution data from LLMs to train, each cluster is posited to embody a distinct semantic topic within the dataset. From these clusters, we extract samples guided by PPL score ranking and predetermined sample thresholds.

Specifically, we utilize the mean embeddings from the last hidden layer of corresponding LLMs. PPL scores are acknowledged as a robust indicator of an LLM's prediction uncertainty concerning specific data samples. Drawing inspiration from Curriculum Learning principles [3], we strategically select data samples within the range constructed by defined sampling thresholds to further enrich data diversity, concurrently mitigating the inclusion of outlier representations. The sampling thresholds are set as (25%, 75%) and (n, k) combination is set as (30, 100). The complete process of first stage is described in Algorithm 1.

3.2 Dynamic Data Sampling

In this stage, we iteratively generate bunches and dynamically adjust target counts for each bunch to select the most representative data samples while keeping diversity from $\mathcal{D}'$, drawing inspiration from [42,12].

The iterative selection for i-th sample in j-th bunch is defined as follows:

$$x_k = \arg\max \left(\sum_{d \in \mathcal{S}_j^{k-1}} C_1(x_k) - \sum_{d \in \mathcal{D}' \setminus \mathcal{S}_1 \setminus \mathcal{S}_2 \setminus \cdots \setminus \mathcal{S}_j^{k-1}} C_2(x_k) \right) \tag{1}$$

where $\mathcal{D}' \setminus \mathcal{S}_1 \setminus \mathcal{S}_2 \setminus \cdots \setminus \mathcal{S}_j^{k-1}$ denotes the remaining part of the data in $\mathcal{D}'$ after selecting $k-1$ samples in j-th bunch, $C_1(x_k)$ and $C_2(x_k)$ are constructed by

maximizing submodular gains $P(x_k)$ in feature space defined in GraphCut [12]:

$$P(x_k) = \sum_{d \in \mathcal{S}_1^{k-1}} \underbrace{||f(d) - f(x_k)||^2}_{C_1(x_k)} - \sum_{d \in \mathcal{D}' \setminus \mathcal{S}_1^{k-1}} \underbrace{||f(d) - f(x_k)||^2}_{C_2(x_k)} \tag{2}$$

where $\mathcal{S}_1^{k-1}$ denotes selected examples and $\mathcal{D}' \setminus \mathcal{S}_1^{k-1}$ denotes the remaining examples. Moreover, the target sample size t_i for the i-th bunch is calculated to ensure a balanced representation of the dataset across the bunches, taking into account their respective densities, which is calculated as:

$$t_i = \max\left(\text{int}(\frac{|\mathcal{S}_i|}{m} \times p), 1\right) \tag{3}$$

where $|\mathcal{S}_i|$ represents the size of a certain bunch, m is the summation of the sizes of all bunches, and p is the final number of samples computed as the product of $|\mathcal{D}'|$ and sampling ratio. This step guarantees that no bunch is over- or underrepresented in the final result dataset $\mathcal{D}_f$. We set r as 10% and b as 30 for default settings. The complete process of Dynamic Data Sampling is described in Algorithm 2.

4 Experiments

4.1 Experimental Setup

Models and Datasets We select three open-source LLMs in our experiment: LlaMA-7B [28], ChatGLM3-6B [8], and Baichuan2-7B [2]. Moreover, we select **11** representative instruction tuning datasets of different scales and construction sources to validate the effectiveness of our method, wich more details provided in Appendix A.

Evaluation Metrics and Baselins The challenge of evaluating instruction-tuned LLMs is widely recognized [4], prompting us to execute extensive experiments to demonstrate the effectiveness and reliability of Bread. In order to perform a reasonable assessment of LLMs across standard tasks and those unseen tasks, we assess them on **established benchmarks** including **MMLU** [9], **Hellaswag** [35], and **OpenbookQA** [20]. Bread is compared against 3 distinct scenarios for thorough evaluation: (1) **Random Selection** utilizes randomly chosen data samples as a baseline for comparison; (2) **Instruction Data Mining Strategies** including Alpagasus [5], Cherry [17], and DQ [42]; (3) **Full training** employs the entire dataset for model training which is a common practice in IT.

Implementation Details We employ LLaMA-Factory framework [4] for IT. In our experiments, we utilize LoRA [10] for parameter-efficient fine-tuning. All training runs are performed on a single NVIDIA RTX 3090 GPU. The LoRA rank is set

[4] https://github.com/hiyouga/LLaMA-Factory

Table 1. Main results of accuracy on MMLU in setting of data keeping ratio as 10%. The scores are averaged across all parts in MMLU: STEM, Social Sciences, Humanities, and Other. Each dataset abbreviation in the table matches the respective dataset in Table 4. Some items are missing as their methods weren't evaluated on those datasets. *Random* denotes uniformly sampling 10% data from the training set.

LLM	Method	Training Dataset										
		Dolly	LIMA	Alpaca	Alpaca-g	Unn-g	Dynosaur	wi-v1	wi-v2	dro-v1	dro-v2	longform
LlaMA-7B	Alpagasus	35.26	–	33.29	–	–	–	–	–	–	–	–
	Cherry	–	–	28.85	–	–	–	26.10	–	–	–	–
	DQ	33.43	34.47	36.29	34.49	34.59	34.40	32.20	34.08	30.75	33.78	32.55
	Bread	**36.11**	**34.62**	**36.53**	**36.94**	**34.74**	**35.99**	34.00	**35.16**	**31.76**	**35.18**	**33.46**
	Full	33.08	31.08	33.16	33.58	32.56	34.48	**34.20**	33.49	30.78	32.76	31.58
	Random	32.58	30.54	32.48	31.96	32.20	33.47	30.89	32.59	28.80	27.60	30.56
ChatGLM3-6B	Alpagasus	**49.38**	–	49.60	–	–	–	–	–	–	–	–
	Cherry	–	–	48.52	–	–	–	48.90	–	–	–	–
	DQ	48.52	49.56	49.65	49.36	48.80	48.73	49.37	47.05	42.91	41.56	45.80
	Bread	49.04	**49.80**	**49.92**	**49.68**	**49.43**	**49.50**	**49.94**	**47.67**	**43.40**	**42.60**	**45.89**
	Full	47.52	48.30	47.26	43.78	45.60	42.60	47.79	44.38	42.78	40.79	44.26
	Random	44.58	47.26	42.38	43.56	44.60	43.20	46.78	44.79	40.60	40.38	41.29
Baichuan2-7B	Alpagasus	**49.29**	–	49.01	–	–	–	–	–	–	–	–
	Cherry	–	–	44.84	–	–	–	48.55	–	–	–	–
	DQ	46.15	28.48	48.77	49.17	28.47	41.88	49.04	47.56	36.08	33.65	30.25
	Bread	47.99	**29.42**	**49.24**	**49.24**	**28.57**	**49.31**	**50.03**	**48.09**	**36.89**	**34.27**	**36.68**
	Full	44.82	28.46	46.74	47.82	27.82	39.15	46.89	44.28	35.96	32.60	29.82
	Random	44.67	24.60	41.68	47.45	26.45	38.26	44.26	43.66	35.88	31.26	28.69

as 8 and the dropout rate is set as 0.1. Given the nature of the datasets under consideration, we limit the maximum input length to 512 tokens. We endeavor to align our training hyperparameters with those documented in prior studies [27], including a learning rate of 5×10^{-5}, a batch size of 2, and the employment of gradient accumulation with a step size of 2 across three epochs. In addition, we implement a Cosine learning rate scheduler without the inclusion of warm-up steps and enable mixed precision training (fp16) to enhance training efficiency and stability.

4.2 Experimental Results

Main Results The results on established benchmarks under data keeping ratio of 10% are shown in Table 1 and Table 2. In most instances, training models with the full dataset does not yield better outcomes than using strategic data selection methods, underscoring that the full dataset contains some low-quality data and quality of data is more crucial than quantity. Furthermore, Bread consistently outperforms all other baseline methods and random selection in most situations across all three benchmarks and three LLMs, showcasing our method's success in preserving both the diversity and the representation of the data.

Ablation Study As shown in Table 3, the omission of either Stage 1 or Stage 2 results in a huge decline in performance. For example, removing Stage 2 of

Table 2. Main results of accuracy on Hellaswag and OpenbookQA in setting of data keeping ratio as 10% and training dataset as Alpaca.

LLM	Method	Hellaswag		OpenbookQA
		In-domain	Zero-shot	
LlaMA-7B	Alpagasus	12.02	12.49	26.00
	Cherry	12.92	12.02	23.60
	DQ	14.38	14.30	25.80
	Bread	**16.44**	**16.81**	**26.40**
	Full	14.02	13.88	22.40
	Random	11.58	11.74	21.56
ChatGLM3-6B	Alpagasus	18.32	17.04	40.60
	Cherry	19.90	19.20	36.60
	DQ	21.82	21.66	40.40
	Bread	**22.58**	**22.10**	**43.00**
	Full	16.38	21.68	38.20
	Random	15.78	21.04	37.29
Baichuan2-7B	Alpagasus	24.20	23.92	39.00
	Cherry	25.17	24.26	42.80
	DQ	25.25	25.85	47.60
	Bread	**25.69**	**26.09**	**51.00**
	Full	24.18	24.02	39.20
	Random	22.06	23.58	40.06

Table 3. Ablation study on Bread's two stages. We select Dolly as training dataset and set the data keeping ratio as 10%.

LLM	Method	MMLU	Hellaswag		OpenbookQA
			In-domain	Zero-shot	
LlaMA-7B	Bread	**36.11**	**16.04**	**16.16**	**25.86**
	Bread *w/o Stage 2*	29.94 (-6.17)	10.88 (-5.16)	12.43 (-3.73)	20.66 (-5.20)
	Bread *w/o Stage 1*	29.46 (-6.65)	10.06 (-5.98)	10.41 (-5.75)	21.36 (-4.50)
ChatGLM3-6B	Bread	**49.04**	**21.76**	**21.88**	**41.26**
	Bread *w/o Stage 2*	48.64 (-0.40)	21.08 (-0.68)	18.66 (-3.22)	38.56 (-2.70)
	Bread *w/o Stage 1*	48.62 (-0.42)	19.85 (-1.91)	18.06 (-3.82)	38.04 (-3.22)
Baichuan2-7B	Bread	**47.99**	**23.86**	**25.87**	**50.60**
	Bread *w/o Stage 2*	46.27 (-1.72)	21.46 (-2.40)	24.38 (-1.49)	47.21 (-3.39)
	Bread *w/o Stage 1*	46.69 (-1.30)	21.58 (-2.28)	24.51 (-1.36)	48.06 (-2.54)

Bread causes a performance drop up to 6.17 for LlaMA-7B. These results underscore that both stages of Bread are crucial for achieving optimal performance across various LLM architectures and benchmarks. The consistent significantly better results with both stages intact across all models and tasks demonstrates the effectiveness of this comprehensive approach in preserving data diversity

and representation, thereby enhancing model performance in complex language understanding and reasoning tasks.

5 Conclusion

This paper introduces Bread, a novel instruction data mining method for enhancing Instruction Tuning with high-quality datasets, which combines Diverse Data Retrieval for data diversity and Dynamic Data Sampling for representativeness. Our empirical results indicate that Bread outperforms previous works, demonstrating its effectiveness across various instruction tuning datasets from different sources and compatibility with multiple LLMs. Therefore, Bread provides a natural and direct way for efficient Instruction Tuning with lower computation costs. In conclusion, Bread not only serves to Instruction Tuning workflows but also offers a potential reduction in environmental impact due to its computational efficiency.

References

1. Agrawal, S., Zhou, C., Lewis, M., Zettlemoyer, L., Ghazvininejad, M.: In-context examples selection for machine translation. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 8857–8873 (2023)
2. Baichuan: Baichuan 2: Open large-scale language models. arXiv preprint arXiv:2309.10305 (2023), https://arxiv.org/abs/2309.10305
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning. pp. 41–48 (2009)
4. Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., Yang, L., Yi, X., Wang, C., Wang, Y., et al.: A survey on evaluation of large language models. arXiv preprint arXiv:2307.03109 (2023)
5. Chen, L., Li, S., Yan, J., Wang, H., Gunaratna, K., Yadav, V., Tang, Z., Srinivasan, V., Zhou, T., Huang, H., Jin, H.: Alpagasus: Training a better alpaca with fewer data. In: The Twelfth International Conference on Learning Representations (2024)
6. Chu, X., Ilyas, I.F., Krishnan, S., Wang, J.: Data cleaning: Overview and emerging challenges. In: Proceedings of the 2016 international conference on management of data. pp. 2201–2206 (2016)
7. Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., Xin, R.: Free dolly: Introducing the world's first truly open instruction-tuned llm (2023), https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm
8. Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., Tang, J.: Glm: General language model pretraining with autoregressive blank infilling. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 320–335 (2022)
9. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300 (2020)

10. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2021)
11. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv preprint arXiv:2311.05232 (2023)
12. Iyer, R., Khargoankar, N., Bilmes, J., Asanani, H.: Submodular combinatorial information measures with applications in machine learning. In: Algorithmic Learning Theory. pp. 722–754. PMLR (2021)
13. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters **31**(8), 651–666 (2010)
14. Jha, A., Havens, S., Dohmann, J., Trott, A., Portes, J.: LIMIT: Less is more for instruction tuning across evaluation paradigms. In: NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following (2023)
15. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Computing Surveys **55**(12), 1–38 (2023)
16. Köksal, A., Schick, T., Korhonen, A., Schütze, H.: Longform: Optimizing instruction tuning for long text generation with corpus extraction. arXiv preprint arXiv:2304.08460 (2023)
17. Li, M., Zhang, Y., Li, Z., Chen, J., Chen, L., Cheng, N., Wang, J., Zhou, T., Xiao, J.: From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. arXiv preprint arXiv:2308.12032 (2023)
18. Lialin, V., Deshpande, V., Rumshisky, A.: Scaling down to scale up: A guide to parameter-efficient fine-tuning. arXiv preprint arXiv:2303.15647 (2023)
19. Liu, S., Zhao, S., Jia, C., Zhuang, X., Long, Z., Lan, M.: Bibench: Benchmarking data analysis knowledge of large language models. arXiv preprint arXiv:2401.02982 (2024)
20. Mihaylov, T., Clark, P., Khot, T., Sabharwal, A.: Can a suit of armor conduct electricity? a new dataset for open book question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2381–2391 (2018)
21. Motamedi, M., Sakharnykh, N., Kaldewey, T.: A data-centric approach for training deep neural networks with less data. arXiv preprint arXiv:2110.03613 (2021)
22. Peng, B., Li, C., He, P., Galley, M., Gao, J.: Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277 (2023)
23. Rawte, V., Sheth, A., Das, A.: A survey of hallucination in large foundation models. arXiv preprint arXiv:2309.05922 (2023)
24. Sinaga, K.P., Yang, M.S.: Unsupervised k-means clustering algorithm. IEEE access **8**, 80716–80727 (2020)
25. Song, P.: Lawgpt. https://github.com/pengxiao-song/LaWGPT (2023)
26. Sun, Z., Shen, Y., Zhou, Q., Zhang, H., Chen, Z., Cox, D., Yang, Y., Gan, C.: Principle-driven self-alignment of language models from scratch with minimal human supervision. arXiv preprint arXiv:2305.03047 (2023)
27. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca (2023)
28. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)

29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
30. Wang, H., Liu, C., Xi, N., Qiang, Z., Zhao, S., Qin, B., Liu, T.: Huatuo: Tuning llama model with chinese medical knowledge (2023)
31. Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Naik, A., Ashok, A., Dhanasekaran, A.S., Arunkumar, A., Stap, D., et al.: Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 5085–5109 (2022)
32. Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., Jiang, D.: Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244 (2023)
33. Yang, Z., Zhang, Y., Sui, D., Liu, C., Zhao, J., Liu, K.: Representative demonstration selection for in-context learning with two-stage determinantal point process. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 5443–5456 (2023)
34. Yin, D., Liu, X., Yin, F., Zhong, M., Bansal, H., Han, J., Chang, K.W.: Dynosaur: A dynamic growth paradigm for instruction-tuning data curation. arXiv preprint arXiv:2305.14327 (2023)
35. Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y.: Hellaswag: Can a machine really finish your sentence? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4791–4800 (2019)
36. Zha, D., Bhat, Z.P., Lai, K.H., Yang, F., Hu, X.: Data-centric ai: Perspectives and challenges. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). pp. 945–948. SIAM (2023)
37. Zha, D., Bhat, Z.P., Lai, K.H., Yang, F., Jiang, Z., Zhong, S., Hu, X.: Data-centric artificial intelligence: A survey. arXiv preprint arXiv:2303.10158 (2023)
38. Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., et al.: Instruction tuning for large language models: A survey. arXiv preprint arXiv:2308.10792 (2023)
39. Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al.: Siren's song in the ai ocean: A survey on hallucination in large language models. arXiv preprint arXiv:2309.01219 (2023)
40. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
41. Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., YU, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., Levy, O.: LIMA: Less is more for alignment. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
42. Zhou, D., Wang, K., Gu, J., Peng, X., Lian, D., Zhang, Y., You, Y., Feng, J.: Dataset quantization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17205–17216 (2023)

A Details of Training Datasets

Table 4. Instruction Tuning training datasets in our experiment. *Human* means the dataset is constructed through human effort, *LLM* means the dataset is constructed completely through LLMs, and *Human and LLM* means both sources.

Type	Dataset	Quantity	Source
Human	Dolly	15,011	[7]
	LIMA	1,029	[41]
LLM	Alpaca	52,002	[27]
	Alpaca-gpt4-en	52,002	[22]
	Unnatural-instruction-gpt4	9,000	[22]
	Dynosaur-sub-superni	66,695	[34]
	WizardLM-v1	70,000	[32]
	WizardLM-v2	143,000	[32]
	Dromedary-v1	360,674	[26]
	Dromedary-v2	287,574	[26]
Human and LLM	Longform	2,3652	[16]