MergeIT: From Selection to Merging for Efficient Instruction Tuning

Anonymous ACL submission

Abstract

Instruction tuning is crucial for optimizing Large Language Models (LLMs), as the quality and diversity of instructional data significantly influence model performance. This naturally underscores the importance of an effective and efficient data selection strategy. However, recent mainstream data selection methods typically rely on LLMs to score instruction quality—taking advantage of their capabilities, but at the cost of high computational overhead and reduced data diversity. To address these limitations, in this paper, we propose **MergeIT**, a novel LLM-based **Merging** strategy for better **I**nstruction **T**uning that shifts the focus from selection to synthesis. MergeIT consists of two stages: first, *topic-aware filtering* clusters and refines the dataset, preserving diversity while eliminating redundancy without relying on LLM-based scoring, significantly reducing time and computational cost. Second, *LLM-based merging* synthesizes semantically similar instructions into more informative and compact training data, enhancing data richness while further reducing the size of the dataset. Experimental results demonstrate that MergeIT enables efficient, diverse, and scalable instruction selection and synthesis, establishing LLM-based merging as a promising alternative to prior scoring-based selection methods for instruction tuning.

1 Introduction

Instruction tuning has emerged as a key technique for enhancing the adaptability and performance of Large Language Models (LLMs) (Dubey et al., 2024; Jiang et al., 2023). By fine-tuning LLMs on carefully curated instruction datasets, e.g. `Alpaca_52k` (Taori et al., 2023), a dataset of 52,000 instructions and demonstrations generated by `text-davinci-003` (Brown et al., 2020)., models can generalize across diverse tasks and prompts. However, selecting high-quality instruction data remains a critical challenge, as the choice of data directly affects fine-tuning outcomes.

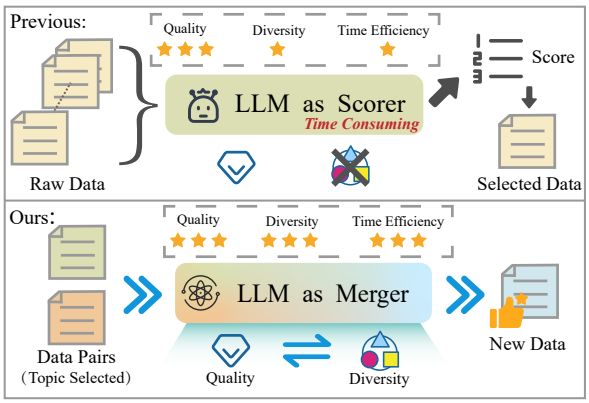

Figure 1: Comparison between our method and prior works. While prior works typically use LLMs as scorers, we instead novelly explore their role as mergers, enhancing diversity and efficiency.

Existing selection methods (Chen et al., 2024) primarily rely on LLMs as instruction quality **scorers**, ranking and filtering instructions based on predefined metrics (Liu et al., 2023; Li et al., 2024b). While this approach retains high-quality instructions, it introduces two major limitations. First, LLM-based scoring is computationally expensive, making large-scale selection infeasible. Second, scoring-based methods often prioritize high-ranked instructions at the expense of diversity, leading to redundant datasets that lack broader generalization. These trade-offs limit the scalability and effectiveness of instruction tuning.

To overcome these challenges, we introduce **MergeIT**, *a novel LLM-based merging strategy for better instruction tuning that moves beyond selection and leverages LLMs as* ***synthesizers*** *rather than mere* ***scorers***. As illustrated in Fig. 1, instead of scoring every single instruction, MergeIT merges semantically related instructions to create more informative, compact, and diverse samples. Comapred to the prior selection-based approaches,

MergeIT improves dataset diversity while reducing time cost.

Particularly, Our approach consists of two core stages: 1) Topic-aware Instruction Filtering – Instead of relying on LLM-based scoring, we first cluster the dataset into semantically meaningful topics and remove redundant instructions within each topic, ensuring diversity while preserving informativeness. 2) LLM-based Instruction Merging – Rather than eliminating similar instructions, we use LLMs to synthesize richer instructions by merging them into a more expressive and information-dense version. This step enhances dataset quality while reducing its size, making fine-tuning both more efficient and effective. By integrating topic-aware filtering with LLM-based merging, MergeIT achieves a faster, stronger, and more diverse instruction selection process, improving both the quality and efficiency of instruction tuning.

Extensive experiments are conducted to validate the effectiveness of MergeIT. Our method achieves state-of-the-art (SOTA) performance across six datasets, demonstrating significant improvements over existing approaches. Notably, our results highlight the feasibility and advantages of leveraging LLMs for instruction merging, a novel direction beyond traditional scoring-based selection. Our main contributions can be summarized as follows,

- We propose **MergeIT**, a novel instruction data optimization framework that shifts from *selection to synthesis*, integrating topic-aware filtering and LLM-based merging to enhance instruction tuning.
- For the first time, we explore the use of LLMs to generate new instructions by merging two similar ones, offering novel insights for the instruction selection and generation.
- Extensive experiments demonstrate the efficacy of our method, achieving high-quality, diverse instruction selection with reduced computational cost.

2 Methodology

Problem Setup: Given an initial instruction dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ from `Alpaca_52k`, where x_i represents the input instruction and y_i denotes the corresponding output, our goal is to select a high-quality subset $\mathcal{D}' \subseteq \mathcal{D}$ for instruction tuning, such that $|\mathcal{D}'| \ll |\mathcal{D}|$.

2.1 Overview

To ensure both diversity and quality in the selected subset while maintaining an efficient selection process, we propose **MergeIT** (**LLM-based Merging Strategy for Better Instruction Tuning**). The overview of our MergeIT method is illustrated in Fig. 3. Our approach consists of two main steps: 1) Topic-aware Filtering; 2) LLM-based Merging.

Given an initial dataset $\mathcal{D}$, the first step filters out redundant examples by selecting only the most informative ones within each topic. This topic-aware strategy naturally ensures that the remaining samples $\mathcal{D}'$ are semantically diverse. Furthermore, by avoiding the use of large LLMs in this stage, we achieve an efficient filtering process. To further compress the data, the second step leverages LLMs to merge similar instances. By harnessing the strong comprehension and generation capabilities of LLMs, we synthesize high-quality and information-rich merged instructions. As a result, the size of $\mathcal{D}'$ is approximately halved, forming the final subset $\hat{\mathcal{D}}$. Unlike prior methods that utilize LLMs solely as scorers , we are the first to explore their potential in a merging framework.

2.2 Topic-aware Filtering

As stated in Sec.2.1, our first step aims to remove redundant examples while preserving the diverse structural semantics of the initial dataset $\mathcal{D}$. To achieve this, our topic-aware filtering first clusters the instructions into specific topics (detailed in Sec.2.2.1) and then selects the most representative subset from each topic, as described in Sec. 2.2.2.

2.2.1 K-means Clustering

We construct the initial clusters of data by introducing K-means algorithm to group pairs according to their instructions. All sentences are represented in embeddings space calculated from `all-MiniLM-L6-v2`, mapping into 384 dimensional features to calculate distances between samples. Given the instruction dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, we first represent each instruction-output pair as a feature vector $f(x, y) \in \mathbb{R}^d$. To partition $\mathcal{D}$ into m topically coherent clusters, we optimize:

$$\min_{\{\mathcal{D}_j\}_{j=1}^m} \sum_{j=1}^{m} \sum_{(x,y)\in\mathcal{D}_j} \|f(x,y) - \mu_j\|_2^2$$
$$\text{s.t.} \quad \bigcup_{j=1}^{m} \mathcal{D}_j = \mathcal{D}$$
$$\mathcal{D}_i \cap \mathcal{D}_j = \emptyset, \quad \forall i \neq j \tag{1}$$

where $\mu_j = \frac{1}{|\mathcal{D}_j|} \sum_{(x,y)\in\mathcal{D}_j} f(x,y)$ is the centroid of topic t_j.

$$t(x,y) = \arg\min_j \|f(x,y) - \mu_j\|_2^2 \tag{2}$$

This provides the topic partitioning $\mathcal{T} = \{t_1, t_2, ..., t_m\}$ that maximizes intra-topic semantic coherence while ensuring clear boundaries between different instruction categories. To validate the feasibility of topic-based partitioning, we visualize the instruction embeddings using t-SNE (van der Maaten and Hinton, 2008) dimensionality reduction in Fig. 2.

2.2.2 Facility Location Function

To control the number of samples, we leverage facility location function as submodular function to select top K representative data in each topic. For each topic cluster $\mathcal{D}_j$, the facility location function is defined as:

$$F(\mathcal{D}'_j) = \sum_{(x,y)\in\mathcal{D}_j} \max_{(x',y')\in\mathcal{D}'_j} \text{sim}(f(x,y), f(x',y')) \tag{3}$$

where $\mathcal{D}'_j \subseteq \mathcal{D}_j$ is the selected subset and $\text{sim}(\cdot,\cdot)$ measures the similarity between two instruction-output pairs in the feature space. This formulation aims to maximize:

$$\max_{\mathcal{D}'_j \subseteq \mathcal{D}_j} \quad F(\mathcal{D}'_j)$$
$$\text{s.t.} \quad |\mathcal{D}'_j| \leq K$$
$$Q(x,y) \geq q_j, \quad \forall (x,y) \in \mathcal{D}'_j \tag{4}$$

The facility location objective ensures each instruction in $\mathcal{D}_j$ is well-related their topics and remains representativeness in the selected subset $\mathcal{D}'_j$.

2.3 LLM-based Merging

Following topic-based alignment and initial subset extraction $\mathcal{D}'_j$, the data volume reduces to approximately 20% of the original corpus, effectively mitigating computational overhead for subsequent LLM processing. However, the resulting subset may still be less effective for LLM fine-tuning due to potential verbosity in remaining samples and quality inconsistencies arising from selection based solely on topical alignment without considering semantic relationships or response quality. To address this, we for the first time propose the LLM-based merging, a cluster-aware methodology that strategically combines semantically related instruction pairs from each topic $\{t_j\}_{j=1}^m$ through third-party LLMs.

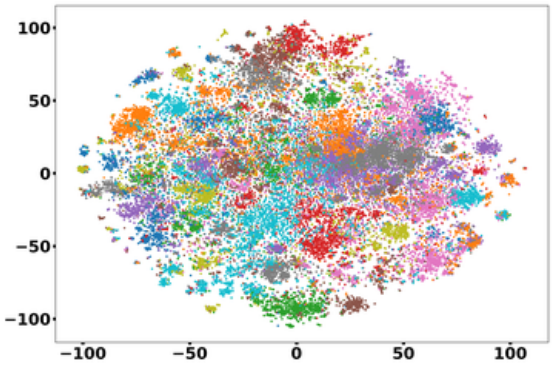

Figure 2: t-SNE visualization of K-means on `Alpaca_52k`. The instructions naturally form distinct clusters, indicating an inherent topical structure effectively captured by clustering.

As illustrated in Fig. 3, starting with the $\mathcal{D}'_j$ from the first filtering stage, for each topic cluster t_j, we establish semantic equivalence classes through:

$$\mathcal{P}_j = \{((x_i,y_i),(x_k,y_k)) \mid \text{sim}(f(x_i), f(x_k)) \geq \tau, (x_i,y_i),(x_k,y_k) \in \mathcal{D}'_j\} \tag{5}$$

where $\mathcal{P}_j$ denotes the set of instruction pairs in topic t_j that exceed the similarity threshold τ, $f(\cdot)$ denotes instruction embedding projection and $\text{sim}(\cdot,\cdot)$ computes cosine similarity. The merging operator $M : \mathcal{D}'_j \times \mathcal{D}'_j \to \hat{\mathcal{D}}$ employs an LLM-based synthesizer:

$$M((x_i,y_i),(x_k,y_k)) = \text{LLM}_{\text{merge}}(\mathbf{c}_{ik}) = (\hat{x},\hat{y}) \tag{6}$$

where $\mathbf{c}_{ik} = [x_i; y_i; x_k; y_k]$ denotes the concatenated instruction-output context. The final training protocol comprises:

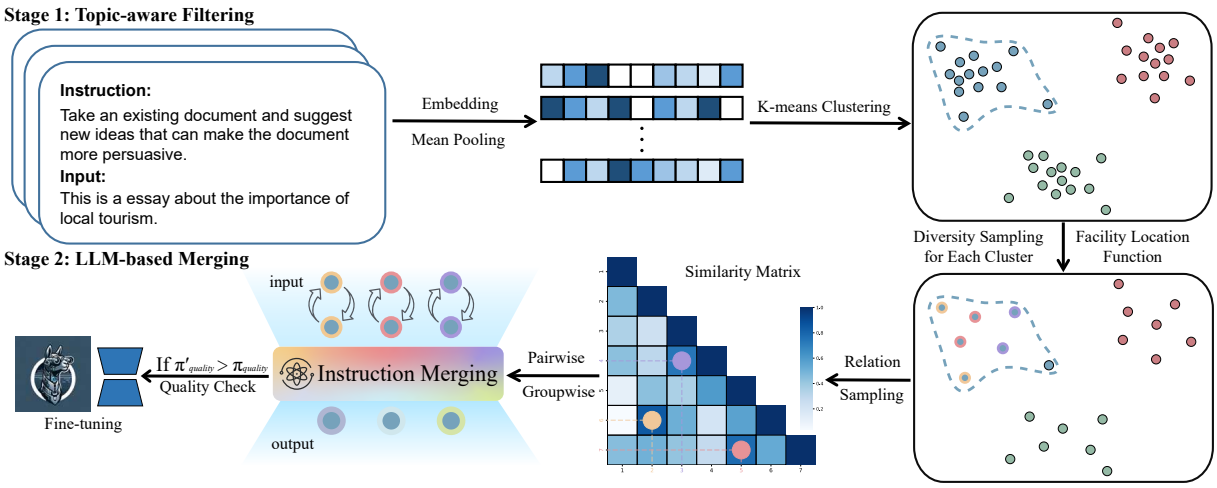

Figure 3: Overview of MergeIT: 1) Topic-aware filtering clusters instructions into topics and filters redundant samples within each topic. 2) LLM-based merging synthesizes new instructions by combining similar pairs.

$$\hat{\mathcal{D}} = \bigcup_{j=1}^{m} \{M(p) \mid p \in \mathcal{P}_j\} \tag{7}$$

$$\mathcal{L}(\theta) = \mathbb{E}_{(\hat{x},\hat{y})\sim\hat{\mathcal{D}}}[-\log p_\theta(\hat{y}|\hat{x})] \tag{8}$$

where $\hat{\mathcal{D}}$ represents the merged instruction set from all topics and $\mathcal{L}(\theta)$ defines the fine-tuning objective for model parameters θ.

Quality Checking. To prevent possible degradation during the merging process, we propose a quality preservation constraint:

$$\pi'_{quality}(M_{i,j}) > \alpha(\pi_{quality}(S_i) + \pi_{quality}(S_j)) \tag{9}$$

where $M_{i,j}$ represents the merged result of samples S_i and S_j, $\pi'_{quality}$ denotes the quality score after merging, $\pi_{quality}$ represents the quality score before merging, and $\alpha \in (0, 1)$ is a parameter controlling the quality threshold, which we set by default at 0.75. This efficient quality checking mechanism ensures that merging operations only proceed when the resultant quality surpasses the weighted quality of the original samples. Our empirical evaluation demonstrates that this quality assessment process incurs negligible time consuming (0.5-1.0 seconds per sample pair), making it particularly suitable for integration into the main merging pipeline without introducing significant processing delays.

This cluster-constrained merging owns two computational advantages. First, it eliminates the $O(n^2)$ pairwise similarity bottleneck by restricting comparisons within pre-clustered topics ($|\mathcal{D}'_j| \ll |\mathcal{D}|$). The localized processing enables: (1) efficient identification of task-specific instruction patterns through intra-cluster analysis, and (2) context-aware merging of logically compatible samples – for instance, consolidating parallelizable tasks into batched instructions or combining complementary reasoning steps into coherent workflows. Second, the synthesis process inherently compresses lexical redundancies through LLM-based paraphrasing, simultaneously enhancing dataset density and instruction quality – a critical improvement over naive concatenation approaches. Additional examples and analysis are provided in Sec. 3.

3 Does Merging Really Make the Data Better for Training?

In this section, we further delve into the real examples of merging in `Alpaca_52k`, to demonstrate the post-processed outcomes, as well as providing analysis to its practical feasibility. As mentioned before, we introduce `Deita` scorer (Liu et al., 2023) as the measurement of quality of training data from the instruction and its response. To clearly reveal the examples of language tasks, we picked three instruction groups (e.g., Translation, Language Analysis, and Calculation), shown in Tab. 1.

To further validate our approach of instruction merging, we explore three key questions: 1) Does instruction merge improve the overall quality? 2) How does the merged data impact the explanation depth of responses? 3) Does LLM effectively integrate knowledge from different components?

Quality Enhancement. Our empirical analy-

Pre-merge Tasks	Post-merge Tasks
Translation Task Group	
Task 1: Instruction: Translate to Spanish Input: "Life is a journey, not a destination." Output: "La vida es un viaje, no un destino." Quality Score: 1.88	**Merged Instruction:** "Translate the given phrase either from Spanish to English or English to Spanish" **Input:** "The sun is not shining today. / La vida es un viaje, no un destino." **Output:** "El sol no brilla hoy. / Life is a journey, not a destination." **Quality Score:** 2.57
Task 2: Instruction: Translate to English Input: "El sol no brilla hoy." Output: "The sun is not shining today." Quality Score: 1.93	
Language Analysis Task Group	
Task 1: Instruction: Categorize part of speech Input: "very" Output: "The word 'very' is an adverb." Quality Score: 1.71	**Merged Instruction:** "Identify the type of sentence structure used and analyze its phrase for the part of speech" **Input:** "The birds are singing, but the sun is hidden on this windy day" **Output:** "The given sentence '...' is a Compound-complex sentence. Also, the phrase 'windy day' within this sentence is an adjective phrase, as it describes the noun 'day' indicating that it is windy." **Quality Score:** 3.13
Task 2: Instruction: Identify sentence structure Input: "The birds are singing, but the sun is hidden." Output: "Compound-complex sentence" Quality Score: 1.23	
Calculation Task Group	
Task 1: Instruction: Identify if the given number is odd or even. Input: "Number: 48" Output: "Even" Quality Score: 2.35	**Merged Instruction:** "Determine if the given number is odd or even, and also check if it is a prime number." **Input:** "Number: 48" **Output:** "The number 48 is even. It is not a prime number because it has divisors other than 1 and itself." **Quality Score:** 3.08
Task 2: Instruction: Is the given number a prime number? Input: "19" Output: "Yes, 19 is a prime number." Quality Score: 2.97	

Table 1: Comparison of Pre-merge and Post-merge Language Tasks

sis reveals a consistent pattern of quality improvement across all examined task groups. As shown in Tab. 1, the quality scores of merged instructions consistently surpass the initial ones. Specifically, in the *Translation Task Group*, we observe an increase from an average score of 1.90 to 2.57, representing a 35.2% improvement. Similar enhancements are evident in *Language Analysis* (141.7% increase) and *Calculation* (15.8% increase) tasks. These substantial improvements in quality scores suggest that our merge strategy effectively combines individual instructions while maintaining coherence.

Response Depth. A notable outcome of our merging process is the improvement in response depth and completeness. Post-merge responses demonstrate a marked increase in explanatory content and reasoning clarity.

For instance, in *Language Analysis Task Group*, since pre-merge instructions are logically coherent with each other, like "categorize part of speech" and "identify sentence structure" all related to the analytical of sentences, LLMs not only connect instructions by simply adding "and" in between, but also emerge to provide more comprehensive outputs. In addition, the Task 1 in the Group is promoted to perform analysis for the whole sentences, rather than merely focusing on a single word, which provides more insights to increment the quality from its original plain requests.

Knowledge Integration. Our analysis further demonstrates the effectiveness of knowledge integration across different task components. The merge process successfully preserves the essential elements of individual tasks while creating cohesive instructions. This is particularly evident in the *Calculation Task Group*, where the merged instruction seamlessly combines number property analysis (odd/even) with prime number verification. The resulting quality score of 3.08 and comprehensive response ("The number 48 is even. It is not a prime number because it has divisors other than 1 and itself") validates that this integration not only maintains but enhances the overall task effectiveness.

4 Experiments

In this section, we evaluate two open-source and commonly used models: `Mistral-7b-v0.3` (Jiang et al., 2023) and `LLaMA3-8b` (Dubey et al., 2024)

with two types of benchmarks: LLM-based (MT-Bench (Zheng et al., 2024), AlpacaEval (Chia et al., 2024)), and Huggingface Open LLM Leaderboard (Hellaswag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), GSM8k (Cobbe et al., 2021), ARC (Clark et al., 2018), and TruthfulQA (Lin et al., 2022)). Several baseline methods, including Alpaca-52k (full dataset), Superfiltering (Li et al., 2024b), Random selection, Perplexity, K-means, and LIMA, are included for comparison.

4.1 Experimental Setup

We adopt LLaMA-Factory[1] as our training base. All fine-tuning experiments utilize LoRA (Hu et al., 2022) with the learning rate 2×10^{-5}, 3 epochs, a batch size of 4, and Cosine scheduler with warmup ratio 0.5. Additionally, we apply 4 pieces of Huawei Ascend 910b 64GB, to train the models. To evaluate Open LLM Leaderboard, we use lm-evaluation-harness[2] as it integrates all of the required benchmarks.

Baselines Setup. For selected baseline models, we extensively experimented with some traditional data selection paradigms. Specifically, K-means baseline clustered the data into 120 groups and selected the most distant samples from each centroid of clusters to realize diversity manner. Perplexity-based method calculates the perplexity score from LLaMa3-8b, formulated below:

$$PP(W) = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 P(w_i|w_1,...,w_{i-1})}, \quad (10)$$

where $PP(W)$ represents the perplexity score for sequence W, N is the sequence length, and $P(w_i|w_1, ..., w_{i-1})$ denotes the conditional probability of predicting the current word w_i given all previous words. For SuperFiltering, we reuse their 10% filtered data from their Instruction-Following Difficulty score.

4.2 Data Scaling

To validate the most applicable number of instruction tuning samples, we implement experiments on testing different data scaling. We include approximately 1k, 6k and 8k of merged data as training subsets and also evaluate on MT-Bench, Hellaswag, ARC and TruthfulQA, as illustrated in Fig. 4. In the given graph, we finalize our corpus size into 6k since it reveals the best trade-off between performance and efficiency, while 9k however demonstrates reverse effects even if the data scales up and 1k data is likely causing the lack of robustness and understanding.

4.3 Main Results

LLM as a Judge. As shown in Table 2, MergeIT-6k achieves the highest performance in LLM scoring, reaching 4.481 and outperforming other baselines by up to 0.518 on Mistral-7b-v0.3 (a 0.842 improvement from base model), while achieving 4.525 on LLaMA3-8b (a 1.107 improvement from base model), surpassing other methods by up to 0.807. Further evaluation on AlpacaEval (Li et al., 2023) in Fig. 5 using GPT-4 shows MergeIT-6k winning in 485 out of 800 comparisons against SuperFiltering-6k's 355 wins, confirming its effectiveness across different judges.

Huggingface Open LLM Leaderboard. Our MergeIT achieves state-of-the-art performance (49.21% average score) across all five tasks, outperforming strong baselines LIMA-6k (47.88%) and K-means-6k (47.78%). The improvements are particularly notable on ARC (54.95%) and TruthfulQA (33.41%). When trained on LLaMA-generated data, MergeIT further obtains 52.96% average score, with significant gains on GSM8k (51.87%) and ARC (54.95%), demonstrating its effectiveness across diverse tasks.

4.4 Ablation Study

To better understand the contribution of each component in our method, we conduct comprehensive ablation studies as shown in Tab. 3 and Tab. 4. Our full model with all three components (6000 samples) achieves the best overall performance with an average accuracy of 52.76% and MT-Bench score of 4.481.

Merging Samples from Given Pairs. We first investigate the impact of our merging strategy. When removing the merging component (12000 samples), the average accuracy drops by 0.84% and MT-Bench score decreases to 4.300, suggesting that our merging strategy effectively enhances model performance by providing more diverse training samples.

Diversity in Topics. The K-means clustering plays a crucial role in maintaining diversity. Without K-means but keeping quality checking (second row), the model's performance drops significantly across all metrics, particularly on GSM8k (-7.01%) and ARC (-3.32%). This indicates that K-means clustering helps ensure a balanced representation of different topics in the training data.

[1] https://github.com/hiyouga/LLaMA-Factory
[2] https://github.com/EleutherAI/lm-evaluation-harness

Model	MT-Bench Score	Huggingface Open LLM Leaderboard (Acc.) ↑					
		Hellaswag	MMLU	GSM8k	ARC	TruthfulQA	Average
Mistral-7b-v0.3	3.639	60.94	58.96	36.62	48.81	22.44	45.55
Alpaca-52k	4.018	61.18	57.73	31.61	53.07	28.76	46.47
SuperFiltering-10%	3.963	60.98	59.34	35.71	49.83	29.99	47.17
Random-6k	4.314	60.83	58.75	35.03	53.07	32.19	47.97
Perplexity-6k	4.352	**61.64**	58.48	37.00	51.88	31.21	48.04
K-means-6k	4.283	60.86	58.45	35.10	52.05	32.46	47.78
LIMA-6k	4.440	60.58	**59.34**	34.34	53.33	31.82	47.88
MergeIT-6k (Ours)	**4.481**	61.40	59.01	**37.30**	**54.95**	**33.41**	**49.21**
LLaMA3-8b	3.418	60.17	**62.13**	50.42	50.26	26.93	49.98
Alpaca-52k	3.718	60.57	61.36	46.10	53.41	30.72	50.43
SuperFiltering-10%	3.968	60.38	**61.95**	50.34	51.54	29.87	50.82
Random-6k	3.912	60.83	58.75	35.03	53.07	32.44	48.02
Perplexity-6k	4.120	61.14	61.09	50.87	53.50	31.33	51.58
K-means-6k	3.731	60.86	58.45	35.10	53.07	32.31	47.96
LIMA-6k	4.450	60.58	61.28	50.34	51.11	34.27	51.51
MergeIT-6k (Ours)	**4.525**	**61.96**	61.49	**51.87**	**54.95**	**34.52**	**52.96**

Table 2: Performance comparison on standard benchmarks (Experiment A). The best results are highlighted in **bold**, and the second-best results are underlined.

Merging	K-means	Quality Checking	# Samples	MT-Bench Score	Hellaswag (Acc.)	MMLU (Acc.)	GSM8k (Acc.)	ARC (Acc.)	μ_{avg}
✓	✓	✓	6000	4.481	61.40	59.01	37.30	53.33	52.76
	✓	✓	12000	4.300	60.31	59.06	34.42	51.37	51.29
✓		✓	6000	4.112	59.68	58.75	30.29	50.01	49.68
✓	✓		6000	4.081	59.81	58.76	33.16	49.56	50.32

Table 3: Ablation studies on different components of our method.

Quality Assurance. Removing the quality-checking component (last row) results in the most significant performance drop, with the MT-Bench score decreasing to 4.081 and average accuracy declining by 2.44%. The impact is particularly pronounced in reasoning-heavy tasks such as ARC (-3.77%) and GSM8K (-4.14%), highlighting the critical role of quality checking in maintaining high-quality training samples and ensuring robust model performance.

Different Merging Strategies. We further compare MergeIT's merging with two alternative strategies in Tab. 4: random selection within topics and simple concatenation-based merging. While random selection preserves topic awareness, it lacks merging, resulting in a performance drop (51.24% average accuracy). Simple concatenation using "and" as the connector performs slightly better (51.56%) but still lags behind MergeIT's merging (52.76%), validating our idea of using LLM-guided merging for generating high-quality samples.

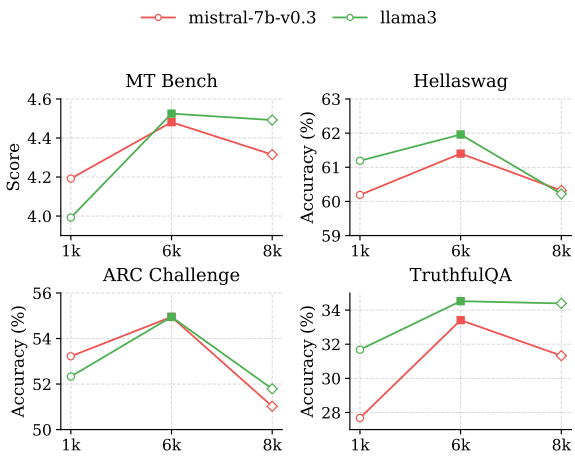

Figure 4: The figure shows the comparison between different scales of number of data in instruction tuning.

Merging Methods	MT-Bench Score	Hellaswag (Acc.)	MMLU (Acc.)	GSM8k (Acc.)	ARC (Acc.)	μ_{avg}
MergeIT Merging - Mistral-7b	4.481	61.40	59.01	37.30	53.33	52.76
Random Select within Topics	4.198	59.98	59.05	35.25	50.68	51.24
Concat Merging	4.200	60.16	58.76	36.47	50.85	51.56

Table 4: Ablation studies on different merging methods.

5 Related Work

5.1 Instruction Data Selection

With the rapid development of large language models and exploration of large-scale training data, research has shifted from model-centric to data-centric approaches. For instance, Zhou et al., 2024 demonstrated that efficient alignment can be performed on small-scale high-quality data. Currently, training data of most large language models (Dubey et al., 2024) must be filtered in advance to ensure high quality, improving training effectiveness and efficiency. For instruction fine-tuning data selection, beyond traditional methods such as coreset selection (Zhang et al., 2024) and clustering (He et al., 2024), recent methods mainly propose new data quality indicators, struggling to balance quality and diversity for high-quality data selection.

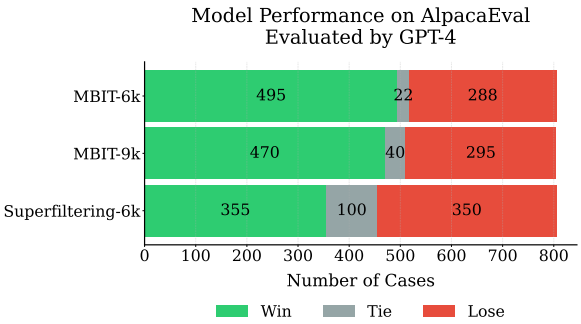

Figure 5: AlpacaEval results. Compared models are MergeIT-6k V.S Alapca-52k full samples (line 1), MergeIT-9k V.S Alapca-52k full samples (line 2) and Superfiltering-6k V.S Alapca-52k full samples (line 3)

5.2 Quality Filtering

The quality of instruction fine-tuning data—such as clarity of instructions and normativity of expressions—directly influences model performance. Common quality filtering methods assign index scores to samples and retain high-scoring ones. Typical indicators include perplexity (Ankner et al., 2024; Mekala et al., 2024), instruction-following difficulty (Li et al., 2024c,b,a), LLM-based scoring (Liu et al., 2023; Song et al., 2024), manual evaluation (Liu et al., 2024a), influence values (Xia et al., 2024; Liu et al., 2024b; Yu et al., 2024), and submodular functions (Agarwal et al., 2025; Renduchintala et al., 2024). While these methods can effectively filter high-quality samples, they do not improve shortcomings in remaining samples or rely solely on LLM-based priors for enhancement, limiting data quality and model improvement. By merging data samples, we leverage information from all samples so they complement each other, maximizing data qualit.

5.3 Diversity-Related Works

Diversity in instruction fine-tuning data—covering domains, formats, and sources—is equally vital. Over-filtering for quality risks omitting task types or domain knowledge, reducing generalization ability. Existing methods often overlook diversity (Shen, 2024) or rely on traditional methods like K-means sampling (Li et al., 2024c; Ge et al., 2024; Maharana et al., 2024) and K-center greedy (Liu et al., 2023; Wang et al., 2024), which often yield suboptimal results. Other strategies remain heuristic, such as the n-gram-based bidirectional graph in Wu et al., 2024. However, these methods discard certain samples, inevitably reducing diversity. In contrast, our instruction merging retains information from all samples, preserving diversity while maintaining quality.

6 Conclusion

In this paper, we introduce MergeIT, a novel framework for efficient instruction tuning. By integrating topic-aware filtering and LLM-based merging, MergeIT effectively filters and combines instruction pairs, reducing dataset size while enhancing informational richness. Notably, we pioneer the use of LLMs for instruction merging, leveraging their generative capabilities to synthesize more informative and compact training data. Our experimental results achieves superior performance over all baselines and demonstrating the potential of LLMs beyond traditional scoring-based selection.

7 Limitation

Our work remains several challenge to be resolved: 1) Our work remains inevitable clustering process to maintain the diversity in the context, which is still served as a less stable method when it comes to larger datasets. 2) Merging process occasionally loses the information given from the samples, which potentially harms the information from the original corpus.

References

Ishika Agarwal, Krishna Killamsetty, Lucian Popa, and Marina Danilevsky. 2025. Delift: Data efficient language model instruction fine tuning. In *International Conference on Learning Representations*.

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L Leavitt, and Mansheej Paul. 2024. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*.

Tom Brown, Benjamin Mann, Nick Ryder, and Subbiah et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2024. InstructEval: Towards holistic evaluation of instruction-tuned large language models. In *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)*, pages 35–64, St. Julian's, Malta. Association for Computational Linguistics.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Mahong Xia, Zhang Li, Boxing Chen, Hao Yang, Bei Li, Tong Xiao, and JingBo Zhu. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 464–478, Miami, Florida, USA. Association for Computational Linguistics.

Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. 2024. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Charlie Bamford, Devendra Singh Chaplot, Diego De Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Leo Raymond Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thomas Lavril, Tao Wang, Thibaut Lacroix, and Wissam El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. 2024a. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16189–16211, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024b. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024c. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7595–7628, Mexico City, Mexico. Association for Computational Linguistics.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and

Tatsunori B. Hashimoto. 2023. 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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Wenxuan Liu, Weiwen Zeng, Kaiyan He, Yijun Jiang, and Jun He. 2023. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Yilun Liu, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, et al. 2024a. Coachlm: Automatic instruction revisions improve the data quality in llm instruction tuning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 5184–5197. IEEE.

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. 2024b. Less is more: Data value estimation for visual instruction tuning. *arXiv preprint arXiv:2403.09559*.

Adyasha Maharana, Jaehong Yoon, Tianlong Chen, and Mohit Bansal. 2024. Adapt- ∞ : Scalable lifelong multimodal instruction tuning via dynamic data selection. *arXiv preprint arXiv:2410.10636*.

Dheeraj Mekala, Alex Nguyen, and Jingbo Shang. 2024. Smaller language models are capable of selecting instruction-tuning training data for larger language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10456–10470, Bangkok, Thailand. Association for Computational Linguistics.

H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Ganesh Ramakrishnan. 2024. SMART: Submodular data mixture strategy for instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12916–12934, Bangkok, Thailand. Association for Computational Linguistics.

Ming Shen. 2024. Rethinking data selection for supervised fine-tuning. *arXiv preprint arXiv:2402.06094*.

Jielin Song, Siyu Liu, Bin Zhu, and Yanghui Rao. 2024. Iterselecttune: An iterative training framework for efficient instruction-tuning data selection. *arXiv preprint arXiv:2410.13464*.

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.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Yejie Wang, Keqing He, Dayuan Fu, Zhuoma Gongque, Heyang Xu, Yanxu Chen, Zhexu Wang, Yujia Fu, Guanting Dong, Muxi Diao, et al. 2024. How do your code llms perform? empowering code instruction tuning with really good data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14027–14043.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Gholamreza Haffari. 2024. The best of both worlds: Bridging quality and diversity in data selection with bipartite graph. *arXiv preprint arXiv:2410.12458*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*.

Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. In *Advances in Neural Information Processing Systems*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. 2024. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data. *arXiv preprint arXiv:2407.15235*.

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

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Merging Examples

We further cherry-pick some merging examples as well as corresponding pre-merge instructions to observe the effects brought by LLMs. More groups and scenarios are provided, shown in Tab. 5

B Small-parameter Models for Merging

Even though MergeIT has saved the most budgets of invoking LLMs API vastly, it still incurs burdens from expensive API calling costs. To solve this challenge, we finetuned a small open-source model, Gemma2-9b from a small portion of data generated by GPT-4o examples. The evaluated results are shown in Tab. 6. From the given table, Gemma2-9b holds same ability even if the size of model is rather smaller than GPT-4o, which proves the process is possible to be migrated to smaller LLMs.

Pre-fusion Tasks	Post-fusion Tasks
Customer Data Management Task Group	
Task 1: Instruction: Update a record in the customer database of a given company according to the instructions Input: "company: ABC Limited\ninstructions: change the contact number for customer 'John Smith'" Output: "The record for customer 'John Smith' has been updated in the customer database of ABC Limited by changing the contact number." Quality Score: 2.31	**Fused Instruction:** "Perform a customer-related task for a given company based on specific instructions." **Input:** "Task: Update contact information\nCompany: ABC Limited\nCustomer: John Smith\nNew Contact Number: 123-456-7890\n\nTask: Send a payment reminder email\nCustomer: John Doe\nAmount Owed: $100" **Output:** "1. The record for customer 'John Smith' has been updated in the customer database of ABC Limited by changing the contact number to 123-456-7890.\n\n2. Email to John Doe:\n\nDear John,\n\nWe hope this finds you well! We noticed that your payment of $100 is past due..." **Quality Score:** 2.89
Task 2: Instruction: Write an email to remind a customer of their past due payment Input: "Name: John Doe, Amount owed: $100" Output: "Dear John,\n\nWe hope this finds you well! We noticed that your payment of $100 is past due..." Quality Score: 2.45	
Conflict Resolution Task Group	
Task 1: Instruction: Provide strategies for personal conflicts Input: "Family dispute situation" Output: "1. Open Communication: Encourage all parties involved to express their thoughts..." Quality Score: 2.43	**Fused Instruction:** "Provide strategies for resolving conflicts in different settings" **Input:** "Family disputes and angry customer emails" **Output:** "To effectively resolve conflicts in both personal and professional settings, consider the following strategies: 1. Open Communication: Encourage all parties involved to express their thoughts..." **Quality Score:** 2.95
Task 2: Instruction: Handle professional conflicts Input: "Angry customer email" Output: "1. Acknowledge the issue 2. Express empathy..." Quality Score: 2.51	
Technical Support Task Group	
Task 1: Instruction: Detect entities from text Input: "Yesterday afternoon, Amazon Web Services went down..." Output: "Entities detected: Amazon Web Services, US-East-1 data center, Virginia." Quality Score: 2.34	**Fused Instruction:** "Provide troubleshooting guide for technical issues" **Input:** "1. 'An unknown error occurred. Error code: 500.'\n2. 'Your internet connection is slow.'" **Output:** "To troubleshoot the error code 500, which is an internal server error, start by examining the server and application logs..." **Quality Score:** 3.12
Task 2: Instruction: Generate tech support conversation Input: "Customer: I need to reset my password." Output: "Tech Support: No problem! What is the email address you use to login..." Quality Score: 2.41	

Table 5: Comparison of Pre-fusion and Post-fusion Language Tasks

Merging Methods	MT-Bench Score	Hellaswag (Acc.)	MMLU (Acc.)	GSM8k (Acc.)	ARC (Acc.)	μ_{avg}
GPT-4o merging	4.481	61.40	59.01	37.30	53.33	52.76
Gemma2-9b merging	4.370	59.05	60.98	36.25	50.68	51.74
Concat Merging	4.200	60.16	58.76	36.47	50.85	51.56

Table 6: Merging methods comparisons. All generated subset of data are trained on Mistral-7b-v0.3.