An Empirical Evaluation of Data Selection Strategies for Instruction Tuning

Anonymous ACL submission

Abstract

Instruction tuning has become a key paradigm for aligning large language models (LLMs), yet the role of data selection remains underexplored compared to model and data scaling. In this work, we move beyond simple scaling to conduct a systematic study of multi-dimensional data selection. We propose a unified framework that balances three pillars: quality, diversity, and complexity. We introduce the Conversation Quality Index (CQI), a scalable composite metric that integrates knowledge depth, topical coherence, and overall content quality. To guarantee broad semantic coverage, we further apply embedding-based clustering over the instruction space. Instruction complexity is quantified using the IC-IFD metric, enabling a structured assessment of cognitive and informational demands. To evaluate these strategies, we construct a task-specific benchmark of 1,346 curated prompts and propose a meta-prompting evaluation framework that generates tailored criteria for each task. Our experiments on the Llama families show that selective sampling consistently outperforms baseline instruction tuning. Combining quality, diversity, and complexity yields the strongest improvements, particularly for larger models. Our findings demonstrate that principled data curation can improve performance and efficiency, highlighting the importance of data-centric approaches for LLM alignment.

1 Introduction

Instruction tuning has become a central paradigm for aligning large language models (LLMs) with user intent. While scaling model size and training data has driven substantial performance gains, recent evidence suggests that **data quality and selection strategy** can be as important as sheer volume. In practice, large instruction datasets often contain redundancy, noisy or low-quality responses, and imbalanced task distributions. Training on such data increases computational cost and may dilute model performance.

A growing body of work has therefore explored **automatic data selection** for instruction tuning, leveraging signals such as difficulty, quality, and diversity. However, prior approaches typically emphasize a single criterion or treat these dimensions independently. Moreover, many evaluation protocols rely on generic scoring rubrics that fail to capture task-specific requirements, limiting the interpretability of improvements achieved through data filtering.

In this paper, we present a comprehensive empirical study of **quality-, diversity-, and complexity-aware data selection for supervised fine-tuning (SFT)**. Our approach integrates three complementary dimensions:

- **Quality**, measured through a multi-task classifier that predicts Knowledgeability, Topic Consistency, and Content Quality, aggregated into a unified Conversation Quality Index (CQI).
- **Diversity**, ensured via large-scale embedding-based clustering combined with task-aware labeling to guarantee broad semantic and functional coverage.
- **Complexity**, quantified using the IC-IFD metric to capture instruction difficulty and reasoning demand.

By combining scalable automatic scoring with clustering-based filtering, we construct compact yet representative training subsets from a 9M-sample corpus. Unlike heuristic or purely random sampling, our framework systematically balances semantic fidelity, representational coverage, and cognitive difficulty.

To rigorously assess the impact of these selection strategies, we introduce a **task-specific evaluation**

benchmark comprising 1,346 curated prompts across nine task categories. Rather than employing a uniform evaluation rubric, we adopt a meta-prompting framework that generates tailored evaluation criteria for each task instance. This enables fine-grained assessment of instruction adherence, factual correctness, completeness, coherence, and safety in a context-sensitive manner.

Our experiments on Llama-3.2-3B and Llama-3.1-8B demonstrate that selective sampling consistently outperforms baseline instruction data. Notably, combining quality with diversity and complexity yields the strongest gains, particularly for larger models, highlighting the importance of structured data curation. These findings reinforce the hypothesis that **careful data selection can rival or exceed gains from scaling data volume alone**.

Our contributions are threefold:

- We propose a scalable, multi-dimensional data selection framework integrating quality, diversity, and complexity signals.
- We introduce a task-specific, meta-evaluated benchmark for more reliable assessment of instruction-tuned models.
- We provide empirical evidence that principled data curation substantially improves SFT efficiency and downstream performance.

Overall, our results give useful guidance on creating better instruction datasets and strengthening data-focused alignment approaches for large language models.

2 Related Works

Recent work has highlighted the importance of data filtering in improving instruction-tuning efficiency. Early approaches relied on large models or reward models to score samples, which is computationally expensive at scale. (Li et al., 2024) introduced Superfiltering, showing that even small models can effectively approximate strong models in ranking examples by instruction-following difficulty (IFD) a score based on perplexity with and without the instruction enabling fast weak-to-strong filtering. (Liu et al., 2023) investigate what makes instruction-tuning data effective, proposing automatic selection methods that score samples along complexity, quality, and diversity. They introduce metrics such as Evol Complexity and Evol Quality, and show that combining these with diversity filtering allows small subsets of data to match or exceed full-data performance, highlighting that careful selection often outweighs sheer data volume. (Burden et al., 2024) propose conversational complexity as a framework to assess large language model (LLM) safety risks, arguing that harmful outputs often arise from multi-turn interactions rather than single prompts. They introduce measures such as conversational length and minimum conversational complexity (an approximation of Kolmogorov complexity) to quantify how much user effort is required to elicit unsafe responses. (Pan et al., 2025) revisit LLM behavior through the lens of compression theory, drawing on Kolmogorov complexity and Shannon information theory. They propose the Syntax-Knowledge model, a hierarchical data-generation framework, to explain how LLMs acquire information from common syntactic patterns toward rarer factual knowledge as model size and training data increase.

(Hui et al., 2024) challenge the prevailing assumption that larger language models (LLMs) are inherently better at evolving diverse and complex instructions. They introduce the Instruction Complex-Aware Instruction Following Data (IC-IFD) metric, which incorporates instruction complexity into the original IFD score, offering a more accurate evaluation of instruction data effectiveness.

(Zhang et al., 2025) propose D3, a data selection framework that jointly optimizes for diversity, difficulty, and dependability. The method formalizes each criterion through dedicated functions, introduces an uncertainty-based prediction difficulty (UPD) score to address context-driven diversity, and incorporates a teacher LLM to evaluate dependability.

In contrast to earlier studies that primarily emphasize instance quality or use diversity filtering as a post-processing step, (Yu et al., 2024) employ k-means clustering to guarantee that the chosen subset is globally representative.

3 Methodology

We employ a two-step method to select a suitable subset for SFT training. In the first step, 500,000 samples are drawn from the 9 million available instances based on a predefined quality metric to prioritize higher-quality data. In the second step, two diversity criteria, domain and overall task type, and

complexity are applied to ensure comprehensive coverage across different conversational contexts.

In the following sections, we describe each step in detail:

3.1 Quality Scoring

Our methodology for assessing conversational quality is organized into three phases: *Data Generation*, *Model Training*, and *Quality Metric*. The first phase produces a reliable annotated dataset, the second introduces a multi-task classification model for automated prediction, and the third consolidates model outputs into a standardized quality metric. Together, these phases balance scalable data creation, robust model generalization, and interpretable evaluation.

3.1.1 Dataset Construction

To train and evaluate the classifier, we constructed a labeled dataset of $10k$ conversations. Instead of relying on costly and time-consuming human annotation, we employed GPT-5-mini as an automatic evaluator.

A structured prompt (see Appendix A) was designed to score conversations along three dimensions:

- **Knowledgeability**: accuracy and depth of conveyed information.
- **Topic Consistency**: degree of coherence and topical relevance throughout the dialogue.
- **Content Quality**: clarity, usefulness, and overall interaction value, while penalizing hallucinations or harmful responses.

Each dimension was rated on a three-point ordinal scale $(0, 1, 2)$, with explicit definitions for each level. To further discourage misleading or low-quality responses, we introduced *negative factor rules* that apply penalties without producing negative overall scores. This procedure yields a multi-task supervision signal, assigning each conversation a triplet of labels (y_1, y_2, y_3).

3.1.2 Model Architecture

We implemented a multi-task classification model based on mmBERT (Marone et al., 2025), a multilingual Transformer capable of handling extended conversational contexts. The architecture comprises:

1. **Shared Encoder**: mmBERT encodes the conversation into contextualized embeddings.
2. **Pooling Layer**: Mean-pooling over the final hidden states (weighted by the attention mask) yields a fixed-length representation.
3. **Task-Specific Heads**: Each evaluation dimension is predicted by an independent linear classification head producing logits over three classes.
4. **Loss Function**: A multi-task cross-entropy objective with class-specific weights to address imbalance. The final loss is the mean across tasks.

Formally, given an input conversation x, the model outputs logits $\mathbf{z}_t \in \mathbb{R}^3$ for each task $t \in \{1, 2, 3\}$. The training objective is:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \mathrm{CE}(\mathbf{z}_t, y_t), \tag{1}$$

where $T = 3$, y_t is the ground-truth label, and CE denotes cross-entropy loss.

3.1.3 Quality Metric

While the classifier outputs dimension-specific predictions, we introduce an aggregate metric—the *Conversation Quality Index* (CQI)—to facilitate standardized comparisons across conversations. CQI integrates semantic quality, lexical diversity, and conversational engagement, producing a single interpretable score in $[0, 1]$.

Let $(K, T, C) \in 0, 1, 2$ be the predicted scores for *Knowledgeability*, *Topic Consistency*, and *Content Quality*. We first compute a normalized weighted average:

$$Q_{\text{raw}} = \frac{w_K K + w_T T + w_C C}{2}, \tag{2}$$

where the weights $\mathbf{w} = (w_K, w_T, w_C)$ are empirically chosen to emphasize semantic fidelity and coherence. In our setup, $(w_K, w_T, w_C) = (0.1, 0.3, 0.7)$.

To account for surface-level lexical diversity and conversational depth, we define two auxiliary factors:

$$\tau = \min(\mathrm{TTR}, 0.8), \qquad \phi = \min\left(\frac{\text{turns}}{5}, 1\right), \tag{3}$$

where τ is the capped type–token ratio and ϕ is the normalized number of dialogue turns.

The final CQI is computed as:

$$CQI = \beta \cdot Q_{\text{raw}} + \gamma \cdot \tau + (1 - \beta - \gamma) \cdot \phi, \tag{4}$$

with $(\beta, \gamma) = (0.8, 0.1)$ controlling the relative influence of semantic quality, lexical diversity, and conversational engagement.

This formulation (i) emphasizes content-level quality while attenuating artificially inflated scores from short or translation-like utterances, and (ii) provides a unified quality indicator that aligns with human judgments of conversational performance.

3.2 Diversity Modeling

To mitigate the substantial size and redundancy of the dataset, we applied a diversity-oriented filtering procedure. Specifically, we employed a clustering-based approach to group semantically similar data points and subsequently selected representative samples from each cluster. This strategy ensured comprehensive semantic coverage while substantially reducing the size of the dataset, resulting in a marked improvement in training efficiency.

3.2.1 Clustering

Although numerous clustering algorithms, such as DBSCAN(Ester et al., 1996) and HDBSCAN(McInnes et al., 2017), are well-suited for capturing complex data distributions, we found that *k*-means clustering was the most practical choice for our dataset scale. Density-based methods, while powerful, exhibit significantly higher computational complexity (often approaching $\mathcal{O}(n^2)$ in worst-case scenarios), making them computationally prohibitive for large datasets. In contrast, *k*-means offers a more tractable complexity of approximately $\mathcal{O}(nkdi)$, where n is the number of data points, k the number of clusters, d the dimensionality, and i the number of iterations. This favorable trade-off between efficiency and semantic coverage makes *k*-means well-suited to our application. When combined with a quality-based sampling step, this approach yields an optimal and representative subset of instruction data.

To perform clustering, we first generated embeddings for each chat using a large-scale embedding model. A key challenge in this step was the limited context length of most models, which prevented the processing of entire conversations—particularly for lengthy, multi-turn dialogues. To overcome this limitation, we employed `Qwen3-Embedding-4B`, which supports a context length of up to 32k tokens, allowing us to represent long conversations with minimal truncation. The initial clustering process resulted in 1,909 clusters; however, a substantial degree of overlap was observed among them. To mitigate this redundancy, we performed a second-stage clustering to merge semantically similar topics. For this refinement, we experimented with three approaches: *k*-means clustering, HDBSCAN, and clustering guided by a LLM.

First-stage clustering: To determine the optimal number of clusters k for the k-means algorithm, we experimented with multiple values of k. We evaluated each configuration using the silhouette score (Vardakas et al., 2024). Although the silhouette score is not a perfect measure of clustering effectiveness, it provides a useful basis for comparing different k-values and selecting the most appropriate one.

The silhouette score is a commonly used metric for assessing the quality of clustering solutions. For each data point i, the silhouette coefficient $s(i)$ is computed as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where $a(i)$ denotes the mean intra-cluster distance (i.e., the average distance between i and all other points within the same cluster), and $b(i)$ represents the minimum mean distance between i and all points in any other cluster. The overall silhouette score is obtained by averaging $s(i)$ across all data points. Values close to $+1$ indicate that clusters are well separated, values near 0 suggest that clusters overlap, and negative values imply potential misclassification of data points.

After wide-ranging experiments, we observed that the silhouette score is not a suitable metric for our dataset. Due to the large scale of the data and the high similarity among embeddings, the resulting silhouette scores are consistently close to zero, providing limited discriminatory power for evaluating clustering quality. This phenomenon can be attributed to the high dimensionality of the embedding space, where pairwise distances tend to concentrate, reducing the relative difference between intra-cluster and inter-cluster distances. As a result, even well-formed clusters may yield low silhouette values. Additionally, the dense distribution of points in the representation space further diminishes the ability of the silhouette score to capture meaningful cluster separation.

We determined that 2048 would be the optimal number of clusters for our analysis, based on the results of preliminary experiments. However, the kmeans algorithm only successfully populated

1909 of these clusters, with several clusters containing as little as a single data point. This outcome indicated a significant degree of over-segmentation, necessitating a post-hoc merging strategy for highly similar clusters.

To successfully combine these clusters, we first determined the overall theme for each cluster. This was done by randomly selecting three examples from each cluster and using GPT-4o-mini to determine a common theme among them. The exact prompt that was used to facilitate this is provided in Appendix A.

Second-stage clustering: After completing the prevoius phase, we introduced an additional field for each chat, referred to as the "task." To generate this label, we first compiled a dataset comprising 100,000 user prompts, sampled uniformly across clusters. These were then input into GPT-4o-mini, which automatically labeled each instance with one of the following task categories: Information Seeking, Instruction Seeking, Text Composition, Brainstorming, Reasoning, Inquiry, Translation, Summarization, Roleplay/Simulation, Text Styling, Code Assistance, and Other. Next, the labeled dataset was used to train a multilingual BERT classifier that assigned a task label to every conversational turn. The overall task of a conversation was determined as the most frequent task label found in its turns.

3.3 Complexity Metric

To measure the reasoning demand of instruction data, we adopt the Instruction Complexity–Instruction Following Difficulty (IC-IFD) score (Hui et al., 2025)), which extends the original IFD metric by explicitly modeling instruction complexity. This formulation captures how strongly an instruction constrains the response and therefore provides a scalable proxy for cognitive difficulty.

Given a model Θ, an instruction Q, and a response A, the IC-IFD score is defined as:

$$IC\text{-}IFD_{\Theta}(Q, A) = \frac{L_{\Theta}(A \mid Q)}{L_{\Theta}(Q) \cdot L_{\Theta}(A)}, \quad (5)$$

where $L_{\Theta}(\cdot)$ denotes the likelihood under the model. Higher scores indicate that the instruction meaningfully reduces uncertainty in generating the response, reflecting greater reasoning or structured problem-solving requirements.

We compute IC-IFD using a continually trained variant of the base model that was intended for instruction tuning. Scores are first calculated at the turn level and then aggregated to obtain a conversation-level complexity estimate. Complexity-aware filtering is applied after quality and diversity selection, prioritizing samples with higher IC-IFD while preserving balanced coverage across task types. This ensures that the final training subset contains cognitively challenging yet diverse and high-quality examples.

Empirically, incorporating complexity improves generalization and reasoning performance, particularly for larger models, highlighting the importance of difficulty-aware data curation.

3.4 Evaluation Pipeline

Comprehensively evaluating large language models remains challenging due to the diverse range of capabilities that must be assessed. Traditional evaluation methods based on next-token likelihood (Brown et al., 2020) fail to capture real-world performance dimensions such as instruction following, factual accuracy, coherence, and practical utility—capabilities critical to user satisfaction in production deployments.

Recent benchmarks have adopted LLM-as-a-Judge methodologies to enable scalable evaluation of open-ended generation tasks. Notable examples include MT-Bench (Zheng et al., 2023), which employs GPT-4 to assess multi-turn conversations, and AlpacaEval (Dubois et al., 2024), which measures win rates against reference outputs. While these approaches represent significant advances over likelihood-based metrics, they typically employ generic evaluation rubrics applied uniformly across all prompts. Such criteria—often limited to broad dimensions like "helpfulness," "relevance," or "depth"—lack the specificity required to assess task-dependent quality indicators accurately.

3.4.1 Motivation for Task-Specific Evaluation

Generic evaluation criteria often fail to capture task-specific requirements. For example, the prompt "Recommend 12 Korean films suitable for a 10-year-old" involves several verifiable constraints: providing exactly 12 authentic Korean films, ensuring age appropriateness, and maintaining genre diversity. Broad metrics such as "helpfulness" or "accuracy" may overlook these essential aspects of response quality.

This limitation extends beyond recommendation tasks. Mathematical reasoning, creative writing, and code generation each require task-specific eval-

uation criteria that generic rubrics often fail to capture.

3.4.2 Our Approach

We address this limitation through automated generation of task-specific evaluation criteria. For each evaluation prompt, we employ a meta-prompting approach wherein GPT-5-mini analyzes the question and produces tailored evaluation dimensions with concrete, verifiable quality indicators. This approach systematically instantiates generic quality principles—instruction adherence, factual accuracy, completeness, coherence, and safety—into task-specific requirements. Our evaluation framework assigns each response a score on a 10-point scale based on these customized criteria. Scores reflect both adherence to core quality dimensions and satisfaction of task-specific requirements. We constructed an evaluation benchmark of 1,346 samples spanning nine task categories (Information Seeking, Reasoning, Text Composition, Translation, Code Assistance, Brainstorming, Text Styling, Summarization, and Instruction Seeking).

3.5 Dataset Construction

We constructed our evaluation benchmark from a corpus of 100,000 unannotated question-response pairs derived from real user interactions with conversational AI systems. This data provides representative coverage of authentic user needs and interaction patterns across diverse domains and task types.

3.5.1 Task Classification

We first classified each question into task categories using a fine-tuned mBERT classifier (Marone et al., 2025). The taxonomy comprises nine categories:

- **Information Seeking**: Factual, conceptual, or definitional queries
- **Text Composition**: Creative or professional writing tasks
- **Summarization**: Text condensation and abstraction
- **Text Styling**: Rewriting for tone, style, or formality adjustment
- **Translation**: Cross-lingual text conversion
- **Reasoning**: Logical, mathematical, or scientific inference
- **Code Assistance**: Code generation, debugging, or analysis
- **Brainstorming**: Idea generation and creative suggestions
- **Instruction Seeking**: Step-by-step procedural guidance

3.5.2 Multi-Stage Filtering

We applied a rigorous filtering pipeline to ensure benchmark quality and evaluation utility:

Stage 1: Content Safety Screening

We employed a specialized classifier to identify and remove prompts containing inappropriate content, including violence, hate speech, sexual content, and other material unsuitable for general-purpose evaluation.

Stage 2: Quality Assessment

Each prompt was evaluated for:

- **Clarity**: Is the user's request unambiguous and comprehensible?
- **Answerability**: Can the prompt be meaningfully addressed without additional context?
- **Discriminative value**: Does the prompt effectively differentiate model capabilities (e.g., reasoning depth, factual knowledge, creative ability)?
- **Judge confidence**: Can an LLM reliably assess response quality for this prompt?

Prompts failing any criterion were excluded. This stage employed gpt-oss-20b (Agarwal et al., 2025) with a structured assessment protocol

Stage 3: Prompt Refinement

Many prompts were extracted from multi-turn conversations and contained residual context from earlier exchanges. To improve data quality and consistency, we rewrote them as fully self-contained prompts while preserving the original evaluation intent.

Stage 4: Final Validation

All selected samples underwent final validation using gpt-oss-120b(Agarwal et al., 2025) to reverify content appropriateness and benchmark suitability, ensuring consistency with our quality standards. This filtering process yielded 1,346 high-quality evaluation samples distributed across task categories as shown in Table 1.

Category	Count
Information Seeking	594
Reasoning	209
Text Composition	203
Brainstorming	84
Translation	92
Instruction Seeking	73
Text Styling	62
Code Assistance	29
Total	**1,346**

Table 1: Distribution of evaluation samples across task categories

3.5.3 Task-Specific Evaluation Design

Unlike prior work that applies uniform rubrics (Zheng et al., 2023; Dubois et al., 2024), we generate customized evaluation criteria for each benchmark item through a meta-prompting approach. A carefully designed system prompt (see Appendix A) instructs GPT-5-mini to analyze each question and produce tailored evaluation dimensions.

3.5.4 Evaluation Criteria Structure

For each benchmark sample, the generated evaluation prompt specifies:

Core Quality Dimensions (applicable to all tasks):

- **Instruction Following Relevance**: Does the response directly address all explicit and implicit requirements?
- **Accuracy Factual Soundness**: Are facts, calculations, and domain-specific claims correct?
- **Completeness Depth**: Does the response fully satisfy information needs without critical omissions?
- **Clarity Coherence**: Is the language well-organized, unambiguous, and appropriately structured?
- **Helpfulness Overall Quality**: Does the response provide practical utility to the user?
- **Safety**: Is content appropriate, non-harmful, and suitable for the intended context?

Scoring Rubric

Each evaluation prompt includes a 10-point scoring scale with explicit performance bands:

- 1–2 (Inadequate): Fails to address the prompt, contains critical factual errors, or includes harmful content
- 3–4 (Poor): Addresses the prompt superficially with major deficiencies
- 5–6 (Fair): Minimally adequate but exhibits notable gaps or omissions
- 7–8 (Good): Solid response satisfying most criteria with minor shortcomings
- 9–10 (Excellent): Comprehensively satisfies all criteria with exceptional quality

Crucially, each performance band is instantiated with concrete examples tailored to the specific question, ensuring consistent interpretation across evaluations.

Evaluation Protocol

We employ GPT-5-mini as the judge model. For each model under assessment:

1. Generate responses to all 1,432 benchmark prompts
2. Apply single-wise evaluation using task-specific criteria
3. Obtain structured output: `"reasoning": "<explanation>", "score": <1-10>`
4. Compute aggregate metrics: overall mean, per-category means, score distributions

The judge model outputs both a numerical score and textual reasoning, enabling qualitative analysis of model strengths and weaknesses beyond aggregate statistics.

4 Results

Table 2 summarizes the performance of different data selection strategies on two Llama models: Llama-3.2-3B and Llama-3.1-8B. We report both the number of tokens used and the model performance on our evaluation metric.

Across the board, all selective sampling methods outperform the unprocessed Instruct baseline. The Random-10k selection improves over Instruct by 0.61 points on Llama-3.2-3B and 0.21 points on Llama-3.1-8B, demonstrating that even arbitrary data increases the models' effectiveness.

Focusing on specific data characteristics further enhances performance. The Quality-10k subset

Model (Method)	# Tokens	Llama-3.2-3B	Llama-3.1-8B
Instruct	-	3.01	4.01
Random-10k	5 070 027	3.62	4.22
Diversity-10k	10 814 297	3.45	4.02
Quality-10k	2 354 241	3.78	4.17
Quality-Diversity-Complexity-10k	18 892 147	3.69	**4.34**
Diversity-Complexity-10k	19 717 887	3.74	4.30
Quality-Diversity-10k	6 307 731	**3.88**	4.29
gpt-4o-mini	-	5.73	-
gpt-5-nano	-	7.64	-

Table 2: Performance comparison of different data selection strategies on Llama models.

yields notable improvements, reaching 3.78 and 4.17 on Llama-3.2-3B and Llama-3.1-8B, respectively, suggesting that high-quality samples are more influential than randomly selected data. Similarly, combinations of selection criteria show varying impacts. The Quality-Diversity-10k selection achieves the best score on Llama-3.2-3B (3.88), while the Quality-Diversity-Complexity-10k and Diversity-Complexity-10k subsets provide competitive gains, particularly for Llama-3.1-8B, where the Quality-Diversity-Complexity-10k achieves the highest value of 4.34.

Interestingly, models that explicitly incorporate diversity and complexity—either alone or in combination with quality—tend to perform better on the larger model (Llama-3.1-8B), suggesting that richer and more varied data helps scale model performance. Comparatively, the Llama models remain below the performance of the stronger gpt-4o-mini and gpt-5-nano, which achieve 5.73 and 7.64, respectively, highlighting room for improvement in data selection strategies.

Overall, these results indicate that data selection strategies combining quality, diversity, and complexity can significantly improve model performance, with the best gains observed when quality is explicitly considered alongside diversity.

5 Conclusion

In this paper, we presented a scalable and principled framework for instruction data selection that jointly optimizes quality, diversity, and complexity. Our approach combines automated multi-task quality scoring, embedding-based clustering, and difficulty-aware filtering to construct compact yet representative training subsets from large-scale instruction corpora. In addition, we introduced a task-specific, meta-evaluated benchmark that enables fine-grained and context-sensitive assessment of instruction-following capabilities, addressing limitations of generic evaluation protocols.

Our empirical findings demonstrate that selective data curation consistently improves downstream performance compared to unfiltered or randomly sampled datasets. Quality-driven filtering yields the most reliable gains, while incorporating diversity and complexity further enhances generalization, particularly for larger models. These results suggest that structured and multi-dimensional data selection becomes increasingly important as model capacity grows. Moreover, we observe that carefully selected small subsets can achieve competitive performance relative to significantly larger datasets, highlighting the potential for reducing computational cost and improving training efficiency.

Future work may explore adaptive and dynamic data selection strategies, integration with active learning, and broader evaluation across model architectures and domains. Overall, this study reinforces the growing importance of data-centric approaches in LLM alignment and provides actionable insights for building more efficient and robust instruction-tuned models.

Limitations

First, our evaluation relies on LLM-based judges, which, although scalable, may introduce bias and require further validation against human assessment. Second, while our clustering and scoring methods are efficient, they may still overlook rare or emerging task distributions. Finally, the framework has been evaluated primarily on supervised fine-tuning and may require adaptation for reinforcement learning or continual alignment settings.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

John Burden, Manuel Cebrian, and Jose Hernandez-Orallo. 2024. Conversational complexity for assessing risk in large language models. *arXiv preprint arXiv:2409.01247*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*.

Tingfeng Hui, Lulu Zhao, Guanting Dong, Yaqi Zhang, and Sen Su. 2025. Unlocking the potential of smaller language models as superior instruction evolvers. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 990–1000.

Tingfeng Hui, Lulu Zhao, Guanting Dong, Yaqi Zhang, Hua Zhou, and Sen Su. 2024. Smaller language models are better instruction evolvers. *arXiv preprint arXiv:2412.11231*.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. *arXiv preprint arXiv:2402.00530*.

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

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205.

Zhixuan Pan, Shaowen Wang, and Jian Li. 2025. Understanding llm behaviors via compression: Data generation, knowledge acquisition and scaling laws. *arXiv preprint arXiv:2504.09597*.

Georgios Vardakas, Ioannis Papakostas, and Aristidis Likas. 2024. Deep clustering using the soft silhouette score: Towards compact and well-separated clusters. *Preprint*, arXiv:2402.00608.

Simon Yu, Liangyu Chen, Sara Ahmadian, and Marzieh Fadaee. 2024. Diversify and conquer: Diversity-centric data selection with iterative refinement. *Preprint*, arXiv:2409.11378.

Jia Zhang, Chen-Xi Zhang, Yao Liu, Yi-Xuan Jin, Xiao-Wen Yang, Bo Zheng, Yi Liu, and Lan-Zhe Guo. 2025. D3: Diversity, difficulty, and dependability-aware data selection for sample-efficient llm instruction tuning. *Preprint*, arXiv:2503.11441.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Prompts

Meta Evaluation Prompt

Prompt

You are a Prompt Engineering Expert specializing in creating evaluation prompts for Large Language Models. Your task is to generate only the final evaluation prompt — output nothing else (no explanations, notes, or reasoning).

Guidelines for Creating the Evaluation Prompt:

- **Input**: You will receive a single user question. Analyze it to determine its primary category (e.g., Reasoning, Code Assistant, Brainstorming, Text Composition, Explanation, Summarization, etc.).
- **Important Note**: Do **not** include the actual user question text in your output — only use the placeholder `{{question}}` wherever the question should appear.

Creating Tailored Evaluation Criteria (Detailed Guidance)

In addition to the general criteria below, you must define **3–8 question-specific evaluation criteria** that reflect what an excellent answer to this exact type of question should demonstrate. Be as **concrete and measurable** as possible — avoid vague phrases like "good quality" or "accurate reasoning."
To design strong tailored criteria:

1. **Identify the intent** of the user's question (e.g., solving a problem, generating ideas, explaining a process, writing code, summarizing, etc.).
2. **Determine the key skill or ability** the assistant must show (e.g., logical reasoning, factual precision, creativity, clarity, structured code design, style consistency, etc.).
3. **Define specific expectations** that would separate weak, average, and excellent answers.
4. **Phrase each criterion as a measurable property**, e.g.:

- For reasoning: "Uses correct logical steps and clearly justifies intermediate conclusions."
- For code: "Produces syntactically correct, efficient, and executable code aligned with the user's intent."
- For creative writing: "Maintains consistent tone and emotional depth while adhering to the given theme."
- For explanations: "Breaks complex ideas into understandable parts with appropriate examples or analogies."
- For factual queries: "Uses accurate, verifiable, and current information."

5. Avoid generic wording such as "good," "clear," or "accurate" unless you specify **what exactly** makes it so.

General Evaluation Criteria

Integrate the following **universal criteria** together with your tailored ones:

- **Instruction Following & Relevance**
- **Accuracy & Factual Soundness**
- **Completeness & Depth**
- **Clarity & Coherence**
- **Helpfulness & Overall Quality**
- **Safety**

Tailored Scoring Rubric

Create a **custom 10-point scoring rubric** specifically for the user's question. This rubric must define what score ranges (1–2, 3–4, 5–6, 7–8, 9–10) mean in relation to your defined criteria. Ensure the descriptions are *context-specific* — e.g., what does an "Excellent" or "Poor" answer mean for *this* question type?

Output Evaluation Prompt Format

[System] Please act as an impartial judge and evaluate the quality of the response provided by the AI assistant to the user question displayed below. The user question will also be repeated at the end of this prompt for clarity and consistency.
CRITICAL: Your response MUST be valid JSON only. Do not include any text before or after the JSON object. Do not use markdown code blocks. Output only the raw JSON.
Your evaluation should be based on the provided criteria and the custom scoring guide. Your response must be in this exact format: {{"reasoning":"","score":<integer score from 1 to 10>}}
[User Question] {{question}}
[Evaluation Criteria] Your assessment should consider the following factors:

- [List all criteria here, starting with the general ones followed by the tailored, question-specific ones. Each criterion should be concrete and measurable.]

[Scoring Guide] Please use the following 10-point scale to rate the response. This guide has been specifically tailored for the user's question.

- **Score 1–2 (Inadequate/Harmful):** [Define what a 1–2 score means for this specific question.]
- **Score 3–4 (Poor):** [Define what a 3–4 score means for this specific question.]
- **Score 5–6 (Fair):** [Define what a 5–6 score means for this specific question.]
- **Score 7–8 (Good):** [Define what a 7–8 score means for this specific question.]
- **Score 9–10 (Excellent/Exemplary):** [Define what a 9–10 score means for this specific question.]

[The Start of Assistant's Answer] {{answer}} [The End of Assistant's Answer]
[User Question] {{question}}

REMINDER: Output ONLY valid JSON in the exact format specified. No additional text, explanations, or markdown formatting.

User Question: {user_question}

Quality Score

Prompt

You are an **evaluator** tasked with scoring a chat conversation based on three key criteria: **Knowledgeability**, **Topic Consistency**, and **Content Quality**. Each criterion is evaluated across conversation turns (a turn includes a user's message and the AI's response). Focus on user inputs for Knowledgeability, as they drive the conversation, and consider both user and AI contributions for Content Quality and Topic Consistency.

General Scoring Rules

- Assign an **integer score** between **0** and **2** for each criterion.
- Scores must reflect **clear, distinct levels** without ambiguity.
- **Positive factors** increase the score; **negative factors** reduce it (minimum score is 0).
- Evaluate the **entire conversation**, breaking down by turns where relevant.
- Provide a brief **comment** explaining the score for transparency.

Knowledgeability

Definition: Measures the depth, accuracy, and relevance of the user's questions or inputs across turns, reflecting their understanding or inquiry.

Scoring Levels

- **0 (Low)**: Vague, incorrect, or factually baseless user inputs.
  - *Example*: "What is AI?" with no context or follow-up.
- **1 (Moderate)**: Some correct and relevant inputs but lacking depth or reasoning.
  - *Example*: "How does AI work?" with basic follow-ups.
- **2 (High)**: Consistently accurate, detailed, and well-reasoned inputs.
  - *Example*: "How do transformer models optimize attention mechanisms?"

Negative Factor

- **Incorrect/Misleading Information**: Reduce score by **1** if user inputs are consistently factually incorrect (e.g., "AI is powered by quantum crystals").

Topic Consistency

Definition: Evaluates how well the conversation stays focused on the initial topic introduced by the user. A topic shift is a distinct, unrelated subject sustained in user or AI messages.

Scoring Levels

- **0**: More than 2 distinct topics.
  - *Example*: Starts with AI, shifts to cooking, then to history.
- **1**: Exactly 2 distinct topics.
  - *Example*: AI and data privacy discussed.
- **2**: Only 1 topic throughout.
  - *Example*: Entirely focused on AI model training.

Note: List distinct topics in the comment section. Only count sustained shifts, not brief tangents. Identify if shifts are user-driven.

Content Quality

Definition: Assesses the relevance, clarity, engagement, and helpfulness of the overall conversation, considering both user inputs and AI responses.

Scoring Levels

- **0 (Low)**:
  - Irrelevant, unclear, or confusing interaction.
  - No effort to build value; poor tone or style.
  - *Example*: User asks vague questions; AI gives generic, unhelpful responses.
- **1 (Moderate)**:
  - Some relevance and clarity but shallow or inconsistent.
  - Limited engagement or usefulness.
  - *Example*: User asks basic questions; AI responds correctly but briefly.
- **2 (High)**:
  - Clear, relevant, engaging, and helpful interaction.
  - Well-structured and contextually appropriate.
  - *Example*: User asks detailed questions; AI provides insightful, tailored responses.

Negative Factors

- **Hallucination**: AI provides factually incorrect or unsupported information.
- **Off-Topic/Irrelevant**: AI deviates significantly from the user's topic.
- **Harmful/Biased/NSFW Content**: AI includes inappropriate material.
- **Repetitive/Stilted Responses**: AI uses canned or robotic answers.

Score Reduction: Subtract **1 point** per negative factor (minimum 0).

Comment Section: For each criterion, provide a brief justification for the score, including examples from the conversation and any negative factors applied.

Cluster Merging Prompt

Prompt

Here are 3 chats from the same cluster. Each chat is separated by —.
Your task:
Read the chats. Find the main theme or subject they share. Write a short, clear title in Persian (max 5 words) that best describes the cluster. Avoid vague terms like "General Chat" or "Miscellaneous".

Chats:
chat1: {chat1}
—
chat2: {chat2}
—
chat3: {chat3}
—
Output: <your title>