Take the essence and discard the dross: A Rethinking on Data Selection for Fine-Tuning Large Language Models

Anonymous ACL submission

Abstract

Data selection for fine-tuning Large Language Models (LLMs) aims to select a high-quality subset from a given candidate dataset to train a Pending Fine-tune Model (PFM) into a Selective-Enhanced Model (SEM). It can improve the model performance and accelerate the training process. Although a few surveys have investigated related works of data selection, there is a lack of comprehensive comparison between existing methods due to their various experimental settings. To address this issue, we first propose a three-stage scheme for data selection and comprehensively review existing works according to this scheme. Then, we design a unified comparing method with ratio-based efficiency indicators and ranking-based feasibility indicators to overcome the difficulty of comparing various models with diverse experimental settings. After an in-depth comparative analysis, we find that the more targeted method with data-specific and model-specific quality labels has higher efficiency, but the introduction of additional noise information should be avoided when designing selection algorithms. Finally, we summarize the trends in data selection and highlight the short-term and long-term challenges to guide future research.

1 Introduction

Large language models nowadays can generate natural and authentic human languages and complete many classic NLP challenges (Naveed et al., 2023; Vaswani et al., 2017; Wang et al., 2022; Zhong et al., 2022). Following the knowledge-based pretraining, the user-oriented supervised instruction fine-tuning endows LLMs with the most significant performance rise.

After the success of LIMA (Zhou et al., 2024), data selection has gradually become a research hotspot, which focuses on excavating efficient criteria to select high-quality samples from existing datasets to fine-tune models in downstream tasks

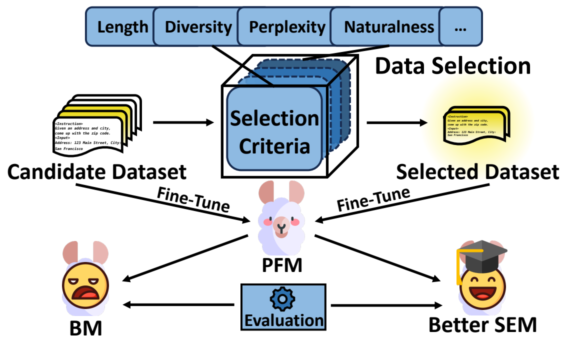

Figure 1: An illustration of data selection for fine-tuning LLMs according to selection criteria. Compared with the Baseline Model (BM), which comes from a Pending Fine-tune Model (PFM) fine-tuned on the full candidate dataset, the Selective-Enhanced Model (SEM) achieves a better performance with less training data.

such as open Q&A and customer service system, as shown in Figure 1. With fewer but better training samples, the selected subset can simultaneously improve fine-tuned LLMs' performance and accelerate their training. Although recent works (Wang et al., 2024; Albalak et al., 2024) have investigated most of the existing data selection methods, there is a lack of in-depth analysis and comparison between each method and a clear development trajectory due to different settings.

To address these issues, we first propose a three-stage data selection scheme that summarizes key parts of the entire data selection process, including data preprocessing, data selector construction, and data selector evaluation. Then, we comprehensively sort out the existing works with the following three aspects: (1) the type of format-conversion of original data after data preprocessing, (2) the information source of quality labels and the corresponding calculation methods used in selector construction, (3) and the various settings in the evaluation process, including candidate datasets, models, and metrics.

To directly compare the existing works, we

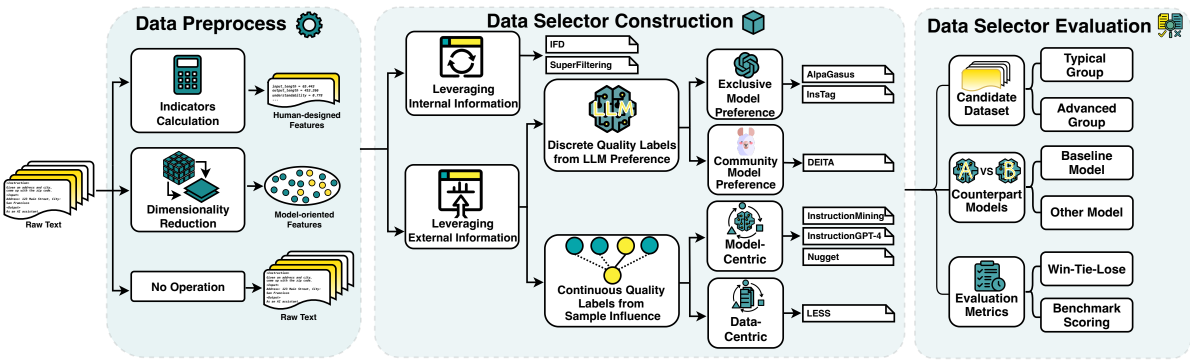

Figure 2: The Three-stage Scheme of Data Selection for Fine-tuning LLMs.

then design ratio-based efficiency indicators and ranking-based feasibility indicators, overcoming their different settings. Specifically, we develop a unified automatic efficiency evaluation method to evaluate them quantitively based on the efficiency curve assumption. We also introduce a manual feasibility evaluation framework that considers simplicity and flexibility to evaluate them qualitatively. After a comprehensive evaluation and analysis of the existing work, we find it difficult to balance efficiency and feasibility in existing works: (1) the more targeted the data selection method, the higher the efficiency, but it also comes with an increase in complexity, descending the feasibility; (2) when adopting more complex algorithms to improve the efficiency of selectors, it is important to avoid introducing additional information as noise to ensure effective selection.

Finally, we not only obtain the technological development trend of existing works from three aspects (Candidate Dataset, Quality Measurement, and Selected Feature) in a timeline but also give the short-term and long-term challenges we need to pay attention to in future research, including the data selection for specific domain and multi-turn conversation and how to find a unified and effective metrics for high-quality data.

2 The Scheme of Data Selection

Data selection for fine-tuning LLMs aims to select a high-quality subset from a given candidate dataset according to some selection criteria, as illustrated in Figure 1. Then, the selected subset is used to fine-tune a vanilla pre-trained language model, which is called a Pending Fine-tune Model (PFM), yielding a Selective-Enhanced Model (SEM). Compared with the Baseline Model (BM) fine-tuned on the full candidate dataset, the SEM is expected to achieve higher performance at a lower cost.

By reviewing existing popular works, we concretize the data selection for fine-tuning LLMs into a three-stage scheme as shown in Figure 2. The scheme consists of (1) data preprocessing (Section 3), (2) data selector construction (Section 4), and (3) data selector evaluation (Section 5) by considering the data features, selection criteria, and usefulness verification respectively.

In data preprocessing, many works retain the original characteristics of the text, while others transform the texts into human-designed features for better explainability (Cao et al., 2023) or model-oriented features for more direct and targeted selection (Xia et al., 2024). After that, data selector construction focuses on the design of selection criteria, which are expected to genuinely reflect the quality of each sample. Existing data selection methods can be first divided up by the information source of quality labels (internal (Li et al., 2024b) or external (Chen et al., 2024)) and then further classified by the different ways of obtaining those quality labels (Liu et al., 2024; Cao et al., 2023; Xia et al., 2024). Finally, data selector evaluation verfies the usefulness of the data selection by the performance improvement of the selective-enhanced model (SEM) over the baseline model (BM). This can be obtained by pairwise comparing the response from the two models directly (Cao et al., 2023) or comparing their scores in some popular benchmarks such as MT-Bench (Lu et al., 2023).

3 Data Preprocessing

While some works preserve the original texts believing that they contain the most information (Li et al., 2024b; Chen et al., 2024), others transform raw texts into representative features. These can be further divided into human-designed features and model-oriented features. The former complies with human instinct, such as the linguistic indicators (Cao et al., 2023; Wei et al., 2023), while the latter is directly extracted from the model itself, such as model gradients (Xia et al., 2024).

Human-designed Features. To guide data selection with respect to human preference, some works use explainable human-designed features with linguistic information. **Instruction-Mining** (Cao et al., 2023) converts a sample into a vector consisting of several NLP metrics, including coherence, naturalness, understandability, etc. Based on this, **InstructionGPT-4** (Wei et al., 2023) [1] additionally introduces the GPT4 score as one of its indicators to better represent the quality of data by measuring whether the generated text adheres to the model's language proficiency.

Model-oriented Features. For more direct and targeted selection, other works use model-oriented features extracted from the model as the representations of data. For example, **LESS** (Xia et al., 2024) creates a datastore of effective and reusable low-dimensional gradient features from the LLM to directly minimize loss on a target task instead of relying on surface form features.

4 Data Selector Construction

Data selector construction focuses on the design of the selection criterion, considering both the information source of the quality label and the way to obtain it, which serves as the fundamental judgment of data quality. The source of quality labels can be divided into internal and external information. The former indicates that the data quality is only related to the information carried by the candidate dataset itself (Li et al., 2024b,a), while the latter considers the information beyond the candidate dataset, such as discrete quality labels from external LLM preference (Chen et al., 2024) and continuous quality labels from sample influence (Li et al., 2024c).

4.1 Leveraging Internal Information

Some works attempt to mine the internal information within the given candidate dataset to obtain quality labels directly. The pioneering work (Li et al., 2024b) proposes Instruction Following Difficulty (**IFD**) as the quality label, which measures the contribution of the instruction to the generation of the corresponding output. To obtain the IFD score, this work first trains an LLaMA-7B (the same as PFM) with a portion of the candidate dataset to be the pre-experienced model. The IFD score is then determined by assessing how the likelihood of generating a specific answer changes when the instruction is provided versus when it is not, using this pre-experienced model.

Inspired by the IFD work, **SuperFiltering** (Li et al., 2024a) adopts a smaller model (GPT-2) as the pre-experienced model to select data by leveraging the consistency in IFD and perplexity from small pre-experienced models to large ones, enabling weak to strong data filtering.

4.2 Leveraging External Information

Other works rely on external information other than the given candidate dataset to obtain the quality of samples, which can be further divided into discrete quality labels and continuous quality labels according to the organizational form.

4.2.1 Discrete Quality Labels from LLM Preference

To reduce the high-cost and time-consuming human annotations of sample quality, some works use exclusive LLMs (such as ChatGPT) or community LLMs (such as LLaMA) to annotate quality automatically, followed by a designed selection algorithm. Such a quality label for the sample is usually discrete and explicit, given by the external LLM with the preference prompt.

Exclusive LLM Preference. One representative work is **AlpaGasus** (Chen et al., 2024), which obtains quality labels by prompting ChatGPT to give each sample a specific score directly and select the samples ranked by the score, align with the what humans would do. The prompt is a designed template with common evaluation aspects, like helpfulness and accuracy, which is universal for any given candidate dataset and PFM. Instead of using a single score in AlpaGasus, **Instag** (Lu et al., 2023) obtains fine-grained quality labels (tags of instruction's intention) annotated by ChatGPT, measuring the quality of samples from both diversity and complexity. Then, they designed a complexity-first diversity sampling algorithm for data selection that takes both perspectives into account.

Community Model Preference. Furthermore, **DEITA** (Liu et al., 2024) utilizes the Evol-Instruct method (Xu et al., 2023) to construct samples of different complexities and qualities for training com-

[1] It also considers CLIP score and multimodal features since it is a multimodal model.

Method	Candidate Datasets	Evaluating SEMs	Counterpart Models		Evaluation Metrics	
			BM	Others	Wins-ties-losses	Benchmark Scoring
AlpaGasus	Alpaca	LLaMA-2 7B	✔	✔	Vicuna, Koala, WizardLM, Self-Instruct	InstructEval
Instruction-Mining	OpenOrca & DOLLY	LLaMA 7B	✔	✔	MT-Bench	OPENLLM, MT-Bench
InstructionGPT-4	MiniGPT-4	LLaMA-2	✔	✘	LLaVA-Bench	MME, VQA, MT-Bench
IFD	Alpaca & WizardLM	LLaMA-2 7B	✔	✘	Vicuna, Koala, WizardLM, Self-Instruct, LIMA	OPENLLM
Superfiltering	Alpaca & Alpaca-GPT4 & WizardLM	LLaMA-2 7B/13B	✔	✘	WizardLM	OPENLLM, AlpacaEval
Nuggets	Alpaca	LLaMA-2 7B	✔	✘	-	MT-Bench, AlpacaEval
LESS	FLAN V2 & CoT & DOLLY & Oasst	LLaMA-2-13B; Mistral 7B	✔	✘	-	MMLU, TYDIQA, BBH
InsTag	WizardLM & UltraChat & ShareGPT	LLaMA-1/-2	✘	✔	-	MT-Bench
DEITA	Alpaca & DOLLY & Oasst & FLAN 2022 & WizardLM & UltraChat & ShareGPT	LLaMA-1/-2 13B; Mistral 7B	✘	✔	-	OPENLLM, MT-Bench

Table 1: The candidate dataset, SEMs, counterpart models, and evaluation metrics used in each method. The "✔" under BM means the work uses the same BM as the SEM; under Others, the "✔" means the work uses models other than BM, including oracle LLMs and other fine-tuned SEMs.

munity models (LLaMA) as the stronger complexity scorer and quality scorer than that trained on the original data. They evaluate the instruction complexity score and response quality score of each candidate sample separately. Then, they designed a score-first, diversity-aware data selection algorithm to select the samples according to the rank of the multiplied score of the two aspects of each sample.

4.2.2 Continuous Quality Labels from Sample Influence

Other works adopt more direct and model-specific methods to select data by utilizing the sample influence on the model's final performance as the quality label, which is usually continuous and implicit. According to the calculation methods of sample influence, they can be further divided into two types: model-centric (Cao et al., 2023; Wei et al., 2023; Li et al., 2024c) and data-centric (Xia et al., 2024).

Model-centric. Instruction-Mining (Cao et al., 2023) employ the Least Squares method to construct the mapping between the 4-dimensional-indicator representations of the sample and the inference loss (Wang et al., 2023; Zheng et al., 2024) on the PFM model. Then, they utilize BLENDSEARCH to select the candidate data effectively, combining global and local optimizations with Bayesian optimization and different local search threads. **InstructionGPT-4** (Wei et al., 2023) adopts the same methodology on a multimodal model by adding visual-caption features. Different from Instruction-Mining, which needs fine-tuning, **Nuggets** (Li et al., 2024c) more directly utilizes the performance difference between taking the sample as the one-shot and the zero-shot setting of PFM on predetermined tasks as the sample influence.

Data-centric. Unlike the above works that attempt to measure the sample influence from the impact on the PFM models performance, **LESS** (Xia et al., 2024) uses the similarity between the gradient of candidate samples and that of the data in existing specific-task datasets to obtain sample influence. They first use the 5% samples of the candidate dataset to warm up the PFM model to obtain the LoRA gradient of each sample, following random projection to get the feature. Then, they design a data selection algorithm, using the average gradient of each task on the validation set as anchor points for similarity calculation with candidate samples' features, and select the top 5% of data points that improved all tasks.

5 Data Selector Evaluation

To evaluate the usefulness of selectors, the method is to select a subset from a candidate dataset through the selector and then fine-tune a model to be the selectively enhanced model (SEM) based on this subset to compare the performance with the same model fine-tuned on full data (Baseline model, BM) or other popular oracle LLMs. Table 1 shows the detailed evaluation setting, including the choice of candidate datasets, counterpart models used in the comparison, and evaluation metrics that provide the performance.

Candidate Datasets. Most of the works (Li et al., 2024a,b; Liu et al., 2024) use the popular open-sourced datasets as candidate datasets to push forward better performance of fine-tuned models by selecting higher-quality samples in them. The candidate dataset is further divided into the typical group, including Alpaca (Taori et al., 2023), Dolly (Conover et al., 2023), FLAN (Wei et al., 2022), etc., and the advanced group developed from the typical datasets to achieve higher quality, including WizardLM (Xu et al., 2023), UltraChat (Ding et al., 2023), etc.

Counterpart Models. To objectively evaluate the performance of the SEM, most works choose BM as the counterpart model for comparison. They tend to use the popular LLaMA series (Chen et al., 2024; Lu et al., 2023) as well as Mistral (Liu et al., 2024; Xia et al., 2024) models as backbones of the SEM and BM to obtain relative improvement evaluation, which directly shows the improvement effect of the selector. Other works (Xia et al., 2024; Chen et al., 2024) compare the SEM with SOTA models (such as GPT-4, Claude, and LLaMA-Chat 7B) to obtain absolute improvement evaluation, which indicates how good SEM achieves.

Evaluation Metrics. Similar to the counterpart models, the evaluation metric adopts the relative and absolute methods to comprehensively evaluate the selector. The absolute metric uses Wins-ties-losses pairing scored by GPT-4 to indicate the direct performance difference between the SEM and counterpart model, while the absolute metric uses benchmark scoring to directly score and rank the SEM. Benchmark scoring is separated into a traditional group examining the loss of the model's response on test datasets (such as MMLU and TYDIQA) and a group using GPT-4 to score on various benchmarks, such as OPENLLM, MT-Bench.

6 Comparing Data Selection Methods

To spot the key factors that lead to superior selectors, we attempt to compare the existing works from the efficiency and feasibility. To address the difficulty of comparison caused by the inconsistency of the evaluation settings (e.g., candidate dataset, PFM, and metrics) across different works, we propose a unified comparison method with several aligned strategies, including developing quantitative ratio-based efficiency indicators and qualitative ranking-based feasibility indicators.

6.1 Efficiency of the Selector

We first compared the efficiency of data selectors in these works, which can measure the accuracy of selectors in selecting the ground-truth high-quality data. The efficiency is mostly expressed by the two indicators in a scatter plot: (1) the performance of SEM and (2) the absolute size of the selected subset. To obtain and unify the efficiency of each work for comparison, we develop two new ratio-based indicators in the efficiency graph with the establishment of the efficiency curve assumption.

Method	SEM	Same Model		Other Models	
		Wins Rate	Bench.	Wins Rate	Bench.
AlpaGasus	LLaMA-2 7B	1.284	0.949	-	-
SuperFiltering	LLaMA-2 7B	1.475	0.962	-	-
InsTag	LLaMA 13B	1.344	-	-	0.985
DEITA	LLaMA-2 13B	1.467	-	-	1.000
InstructionGPT-4	MiniGPT-4	1.443	-	-	-
Nuggets	LLaMA-2 7B	1.519	-	-	-
IFD	LLaMA-2 7B	1.747	-	-	-
LESS	LLaMA-2 13B	(1.570)	0.973	-	-
Instruction-Mining	LLaMA-2 7B	(1.400)	-	0.212	0.991

Table 2: The performance improvement under four evaluation settings. In the Same model, we compare SEM and BM, while in other models, we compare SEM and the same-size models trained based on other backbones (such as LLaMA chat).

6.1.1 Ratio-based Efficiency Indicators

To align original indicators, we develop two ratio-based efficiency indicators: (1) the Performance Improvement Ratio (PIR) and (2) the Selected Dataset Fraction (SDF), which eliminates the bias rooted in the settings (e.g., evaluation metrics and size of dataset).

Performance Improvement Ratio. On the one hand, we design the following steps to obtain the PIR. As outlined in Table 2, we initially categorize evaluation settings into four groups (for more details, see Appendix A.1). We then compute the average performance improvement ratio of the method across various testing datasets under different settings. This is done by averaging the ratios of performance scores of the SEM to those of the counterpart model for each group. Each ratio is calculated by dividing the SEM's performance score by that of the counterpart model in the same evaluation setting. Then, we take the most confident indicator (wins rate in the same model) as the representative indicator for the PIR and estimate missing values by leveraging the consistency of performance under different evaluations in one selector.

Selected Dataset Fraction. On the other hand, we utilize the SDF to assess the impact of data size uniformly. This fraction is calculated by taking the ratio of the selected dataset size to the total size of the original candidate dataset. It ensures that each dataset is represented proportionally, eliminating bias caused by the varying sizes of candidate datasets, which range from 3,439 entries (Wei et al., 2023) to 306,044 entries (Lu et al., 2023).

6.1.2 The Efficiency Curve Assumption

As shown in Figure 3, we draw the unified efficiency graph of each method, where each point is selected based on the best performance reported in

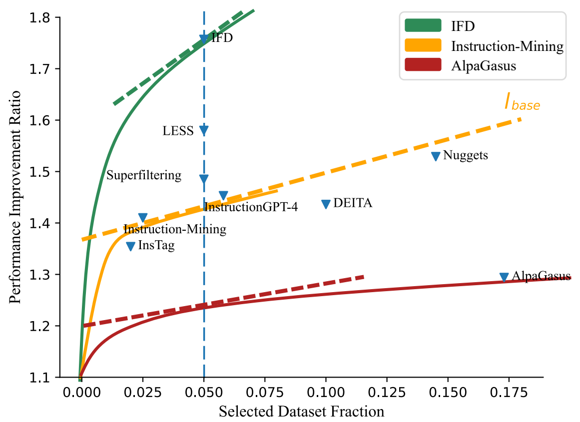

Figure 3: The example of the efficiency curve of three representative works (IFD, Instruction-Mining, and AlpaGasus). The blue dashed line indicates the baselines on the same fraction. Other colored dashed lines indicate the tangent of the curve at a specific fraction (0.05).

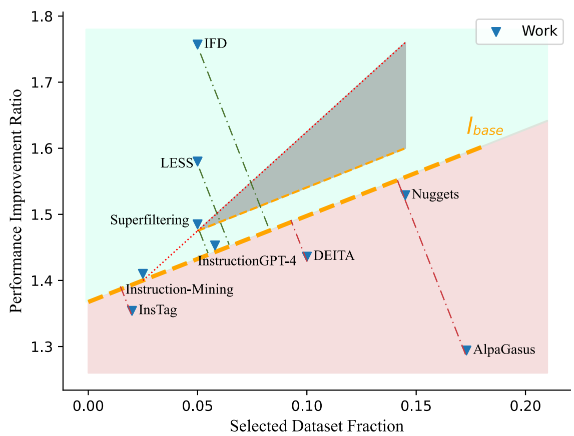

Figure 4: The efficiency graph among popular data selection methods. The yellow line is the efficiency baseline (l_{base}), and the grey area is the infeasible area.

the work. Although a method with a higher PIR and a smaller SDF is more efficient, it is difficult to compare two different works directly as there is no explicit proportional relationship between these two indicators.

Therefore, inspired by the scaling law (Kaplan et al., 2020; Chung et al., 2024) and LIMA (Zhou et al., 2024), we first propose the **efficiency curve assumption**s to get the unified efficiency curve: (1) The unified efficiency curve is logarithm-like, which is upward, concave, and approaching linear after experiencing a rapid but short increase; (2) The slope of the superior efficiency is always larger than the inferior efficiency. (More information is shown in Appendix A.2). Based on this, we first draw the **unified efficiency curve** of each work and select three representative works (IFD, Instruction-Mining, and AlpaGasus) as an example in Figure 3. It can result in a lossless shift of the work along the curve, allowing for the comparison of various works within the same selected dataset fraction.

To compare any two methods directly, we further set up the **efficiency baseline** (l_{base}) by using the approximate estimation of the line connecting Instruction-Mining and Instruction-GPT4, where they are almost on the same efficiency curve due to adopting a similar data selection method, as shown in Figure 4.

After that, we can indicate the superior or inferior efficiency of work (represented as the green or red line) in comparison to the baseline by the signed distance between them, as the distance is with a fixed proportion to the efficiency difference. Methods that perform above the baseline are represented by a green line, indicating superior efficiency compared to the baseline, whereas those below it are shown with a red line, indicating inferior efficiency compared to the baseline. The larger the distance, the greater the efficiency difference. We also observe an infeasible area, but it does not affect our comparative work in this paper. Further details are available in Appendix A.3.

6.2 Feasibility of the Selector

On the other hand, we expect excellent methods to be not only highly efficient but also easy to use in practice. Therefore, we propose Simplicity and Flexibility as two ranking-based feasibility indicators, which qualitatively assess the implementation difficulty and competence in handling new selection tasks of the existing works, as shown in Table 3.

6.2.1 Ranking-based Feasibility Indicators

Simplicity. It evaluates the complexity of the selection process and the reproducibility (Rep.) of work. In terms of complexity, we focus on the number of LLM models that need to be trained during the selection process, as well as the overall algorithm steps (including the number of times using LLM inference). Regarding reproducibility (Rep.), we manually check whether the method released its open-source code that can be easily adapted to other scenarios. More details can be found in the Appendix A.4.1.

Flexibility. It mainly considers whether a data selection method can be more easily applied to other scenarios based on transferability and exten-

Methods	# Trained LLMs	# Algorithm Steps (# Using LLMs)	Rep.	**Simplicity**	Transferability		Extensibility	**Flexibility**	**Feasibility**
					Model Free	Dataset Free	ChatGPT/GPT-4 Free		
AlpaGasus	0	2(1)	✘	**1**	✔	✔	✘	**1**	**1**
InsTag	0	3(1)	✘	**2**	✔	✔	✘	**1**	**2**
Nuggets	0	4(2)	✔	**2**	✘	✔	✔	**2**	**3**
SuperFiltering	1*	3(1*)	✔	**3**	✘	✘	✔	**4**	**4**
IFD	1	3(1)	✔	**4**	✘	✘	✔	**4**	**5**
LESS	1	4(2)	✔	**5**	✘	✘	✔	**4**	**6**
DEITA	2	5(4)	✔	**6**	✔	✘	✘	**3**	**6**
Instruction-Mining	129	4(0)	✘	**8**	✘	✔	✔	**2**	**7**
InstructionGPT-4	30	4(1)	✔	**7**	✘	✔	✘	**5**	**8**

Table 3: The feasibility ranking of existing methods, considering the simplicity and flexibility. The former consists of three indicators: (1) # Trained LLMs, (2) # Algorithm Steps, and (3) Reproducibility, while the latter considers extensibility and transferability. * indicates SuperFiltering trained a GPT-2 instead of LLaMA model. More details are shown in Appendix A.4.

sibility. Transferability mainly depends on whether the construction of the selection method depends on the PFM model (Model Free) and given dataset (Dataset Free), while extensibility mainly considers whether the method relies on certain specific necessary models (such as ChatGPT and GPT-4). More details can be found in the Appendix A.4.2.

Finally, we provide a comprehensive feasibility rank for the existing models, taking into account both simplicity and flexibility, as shown in Table 3. The priority of each element in the simplicity is the number of trained LLMs, algorithm steps, and whether it has reproducibility. The priority of each element in Flexibility is Model Free in Transferability, Dataset Free in Transferability, and ChatGPT/GPT-4 Free in Extensibility.

6.3 Overall Consideration of the Selector

It can be visually seen from Figure 4 the efficiency ranking of each method. The IFD is the best, but AlpaGasus is the worst. This is because IFD is more targeted: The Instruction Following Difficulty score is calculated not only based on information from within the candidate dataset but also on the feature extracted by the pending fine-tuned model (PFM). Moreover, its quality labels come from the loss of the model, which is more direct and does not introduce external information interference. On the other hand, AlpaGasus only utilized external ChatGPT scoring without considering the impact of specific data quality distribution, characteristics of the PFM, and optimization objectives on the performance of the model trained after data selection.

Although the performance is not satisfactory, AlpaGasus performs the best in terms of flexibility, as shown in Table 3. Its high simplicity, due to not requiring training in LLM and having fewer steps in the data selection process, makes it easy for subsequent works to reproduce its results, even though no official publicly available source code exists. In addition, it can be more freely transferred into other scenarios (model-free and dataset-free) as it only relies on ChatGPT without any other information. Although Instruction-Mining and InstructionGPT-4 perform better than AlpaGasus, they sacrifice a lot of feasibility due to their heavy reliance on fine-tuning numerous LLMs and complex quality indicators from the dataset and models.

In summary, existing methods are difficult to achieve both high performance and high feasibility simultaneously. We observe that the more targeted the data selection method, the better the performance of SEM will be. For example, DEITA is more complex than Alpaca due to training the LLM based on PFM and considering the diversity of the data, resulting in better performance. However, more complex processes and algorithms may introduce additional external information than optimize the target directly, and they are also more difficult to transfer. For example, LESS performs worse than IFD, although it is more complex due to the introduction of external datasets.

7 Discussions

7.1 Trends

To explore the current research trends, we have sorted out the existing work from three aspects (Candidate Dataset, Quality Measurement, and Selected Feature) in chronological order, as shown in Figure 5. It is worthwhile to notice that there is a clear trend: current research is gradually evolving toward more targeted data selection.

Specifically, the selector evolves from general to specific in selecting the candidate dataset, where the general one can select any dataset once constructed (Cao et al., 2023), and the specific one has to adjust according to the candidate dataset (Li et al., 2024b). The quality measurement be-

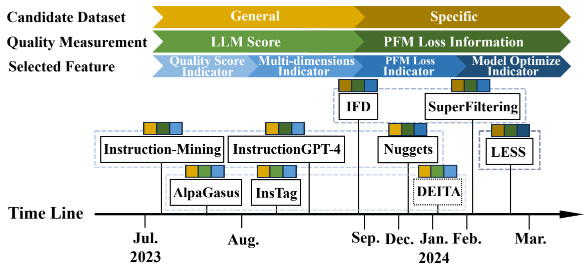

Figure 5: The timeline of the data selection methods.

comes more straightforward to the PFM, which is developed from the LLM score of the external model (Chen et al., 2024) to PFM loss (Xia et al., 2024) coming from the internal model. The selected feature becomes more complex from using concrete indicators (Quality Score and Multi-dimension Indicator) to abstract indicators (PFM Loss and Model Optimize indicator). The concrete indicator introduces semantic factors to explain data quality, while the abstract indicator uses the information from PFM such as loss (Li et al., 2024a) and gradient (Xia et al., 2024).

Additionally, the number of concrete indicators used in the selector increases. DEITA and InsTag employ more indicators than AlpaGasus, which solely relies on quality scores from Chat-GPT. Furthermore, DEITA and InsTag achieve far better overall performance than AlpaGasus due to their taking data diversity into consideration.

7.2 Challenges

Although there has been significant progress in data selection for fine-tuning LLMs, we still face both short-term and long-term challenges.

In the near future, the urgent task is to provide solutions for specific data selection needs, such as specific domains and multi-turns conversations. **Specific domains**. Most data selection methods focus on overall performance improvement, but the contribution of selected data to different domains is not the same. The existing works (Cao et al., 2023; Wei et al., 2023; Chen et al., 2024; Lu et al., 2023; Li et al., 2024c) demonstrated that selected data can bring significant improvements in writing and role-playing but minor improvements in mathematics and reasoning. Although LESS provides a task-oriented data selection method, future work still needs to consider how to dynamically select data based on the model's shortcomings in a specific domain to improve it in specific domains without affecting other domains. **Multi-turn conversations**. Most existing data selection methods are aimed at single-turn conversations because their quality is easier to measure but lacks attention to multi-turn conversation data. Although DEITA (Liu et al., 2024) viewed the multi-turns conversation as multiple single-turn Q&A, they did not consider the characteristics of the multi-turns, such as global goal and consistency in a conversation.

From a long-term research perspective, there are two more in-depth questions that need to be explored: how to balance performance and flexibility and how to find a unified metric for measuring data quality. As we mentioned in the analysis, it is difficult for current research to achieve excellent performance in both efficiency and feasibility because their improvements in efficiency tend to use more refined and targeted methods rather than truly more effective select paradigms. The reason behind this is that existing work considers various indicators to measure data quality for selection but only starts from the external model's observations of the data or the impact of data selection on model performance rather than the quality distribution of the dataset itself. Therefore, exploring a unified and effective metric that can uniformly measure data quality is one of the fundamental issues in the data selection research field.

8 Conclusion

In this paper, we conducted an extensive survey on data selection for fine-tuning large-scale language models. We first construct a three-stage data selection scheme for the entire process and review the current research progress of data selection based on it, including data preprocessing, data selector construction, and data selector evaluation. To address the issue of incompatibility caused by different experimental settings, we propose a unified comparison method from quantitative efficiency evaluation and qualitative feasibility evaluation by designing ratio-based indicators and ranking-based indicators. We find that the data selection methods achieve higher efficiency with data-specific, model-specific, and target-specific designs, but the complex methods could improve efficiency only if it is designed to avoid external information noise. Therefore, it is difficult for the existing methods to balance efficiency and feasibility. Apart from drawing the timeline of the existing work, we also point out the short-term challenges and long-term challenges for future research.

Limitation

We mainly research data selection for instruction fine-tuning LLMs instead of data rewriting or augmentation. Although we have already comprehensively examined the existing works, we acknowledge that there may still be some works we neglected, especially the very recent work that was published on the preprint platforms.

Besides, we focus on outlining the scheme of existing work on data selection and propose an analytical method for comparing various works directly. Therefore, the descriptions of each work could be limited to key points relevant to our study rather than providing a comprehensive overview due to limited space.

References

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2023. Instruction mining: When data mining meets large language model finetuning.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

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

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024c. One-shot learning as instruction data prospector for large language models.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. instag: Instruction tagging for analyzing supervised fine-tuning of large language models.

Elizaveta Moskovskaya, Olesya Chebotareva, Valeria Efimova, and Sergey Muravyov. 2023. Predicting dataset size for neural network fine-tuning with a given quality in object detection task. *Procedia Computer Science*, 229:158–167.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. 2024. A survey on data selection for llm instruction tuning.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Supernaturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. 2023. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning.

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

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.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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 Appendix

A.1 Performance Improvement Ratio

Method	**SEM**	**Same Model**		**Other Models**	
		Wins Rate	**Bench.**	**Wins Rate**	**Bench.**
AlpaGasus	LLaMA-2 7B	1.284	0.949	-	-
SuperFiltering	LLaMA-2 7B	1.475	0.962	-	-
InsTag	LLaMA 13B	1.344	-	-	0.985
DEITA	LLaMA-2 13B	1.467	-	-	1.000
InstructionGPT-4	MiniGPT-4	1.443	-	-	-
Nuggets	LLaMA-2 7B	1.519	-	-	-
IFD	LLaMA-2 7B	1.747	-	-	-
LESS	LLaMA-2 13B	(1.570)	0.973	-	-
Instruction-Mining	LLaMA-2 7B	(1.400)	-	0.212	0.991

Table 4: The performance improvement under four evaluation settings. In the Same model, we compare SEM and BM, while in other models, we compare SEM and the same-size models trained based on other backbones (such as LLaMA chat).

Since different works use different evaluation methods, it is difficult to compare them directly. Therefore, to uniformly evaluate their performance, according to the compared counterpart model, we first divide the various evaluation settings mentioned in all works into BM comparison with SEM on the same PFM and comparison with other models (such as LLaMA chat). Then, we further divide them into wins rate and benchmark improvement (Bench.) with the different metrics. In total, we have four evaluation settings, as shown in Table 4, and we take the average of each type in Eq. (1) if it has multiple evaluations in one setting.

$$\frac{1}{n}\sum_{i=0}^{n}\frac{X_i}{Y_i} \tag{1}$$

where X_i and Y_i are, respectively, the performance of the SEM and the counterpart model under the same evaluation setting i, and n is the total number of the evaluation settings using the same kind of evaluation metric (wins-ties-losses or benchmark scoring) and counterpart model. We then choose the wins rate under BM as the ratio indicator of PIR not only because it directly reflects the improvement effect made by the selector but also because most of the works provide this value.

To fill the missing value, we leverage the consistency of model performance: the same model should perform similarly under different categories. Therefore, we obtain the bridge function by linearly regressing the other works with the wins rate under BM as the label, then use the bridge function to transfer the value of work under other categories into the wins rate under BM.

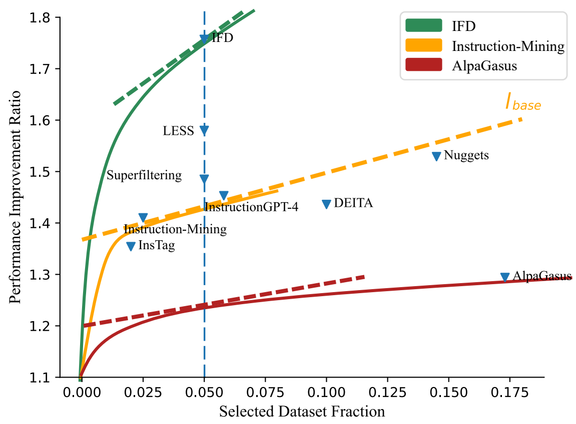

Figure 6: The example of the efficiency curve of three representative works (IFD, Instruction-Mining, and AlpaGasue). The blue dashed line indicates the baselines on the same fraction. Other colored dashed lines indicate the tangent of the curve at a specific fraction (0.05).

A.2 Efficiency Curve Assumption

To directly compare the work with different performance improvement ratios and selected data fractions, we construct an efficiency curve assumption, which consists of two parts:

(1) According to relevant theories (such as scaling law) (Kaplan et al., 2020; Sun et al., 2017; Moskovskaya et al., 2023), they suggest that the impact of logarithmic data size on the loss is linear if the augmented dataset maintains the same quality structure. Inspired by these theories, we account for a dataset with fixed quality, its function of the performance improvement ratio, and selected dataset fraction compilation To the logarithmic-like function, which is upward, considered, and approaching linear after experiencing a rapid but short increase. This way, we can move it on this performance curve to different proportions for easy comparison (such as 0.05, where the blue dash line is located) while maintaining the same efficiency.

(2) The efficiency of the method represented by the above curve with a larger slope is superior to that represented by the below one. It can be intuitively derived from the first part with the fact that high-quality data leads to better performance of SEM cite zhou2024lima Therefore, if the method is superior, which indicates its selected dataset has good quality structure, it increases the performance improvement must be greater than the inferior one at every point of selected dataset fraction.

Therefore, the efficiency (Eff^k) is reflected on the slope of the efficiency curve function ($f(x; Eff^k)$), where the superior efficiency ($Eff_n^s = \frac{df_s(x_n)}{dx_n}$) is larger than the inferior efficiency ($Eff_n^i = \frac{df_i(x_n)}{dx_n}$) at every fraction of selected subset (x_n). Although the slope of the curve is unobtainable because the mathematical expression of the efficiency curve is inaccessible, the comparison efficiency ($CEff$) of work is reflected in its position by leveraging the curve assumption on the baseline. The comparison efficiency transfers the representation of efficiency from the slope of the curve into the distance between the curve and the baseline, which reflects the efficiency difference between them.

As shown in Eq. (2) and (3), for the comparison efficiency of a certain work k (Eff^k) and the baseline (Eff^b), the comparison efficiency ($CEff^k$) can be calculated by the Eq (4).

$$Eff^k = \int \frac{df_k(x)}{dx} dx \tag{2}$$

$$Eff^b = \int \frac{df_b(x)}{dx} dx \tag{3}$$

$$CEff^k = Eff^k - Eff^b = f_k(x) - f_b(x) \tag{4}$$

where x in Eq. (4) is set as same as work k (x_k) which is the only known point of $f_k(x)$. Then, the $CEff^k$ can be represented as Eq. (5):

$$CEff^k = f_k(x_k) - f_b(x_k) \tag{5}$$

To make the comparison efficiency straightforward, we scalar it into the distance between the work k and the baseline Let D_{kb} be the distance between work k and the baseline, and θ be the angle between the baseline and x-axis as shown in Eq. (6).

$$D_{kb} = CEff^k \cos(90^\circ - \theta) \tag{6}$$

As a result, the $CEff^k$ is proportional to D_{kb} as shown in Eq. (7).

$$CEff^k \propto D_{kb} \tag{7}$$

The comparison efficiency method virtually scalars the SDF of all works into an identical value according to the baseline, where the efficiency of work is reflected in the difference between the actual and virtual PIR.

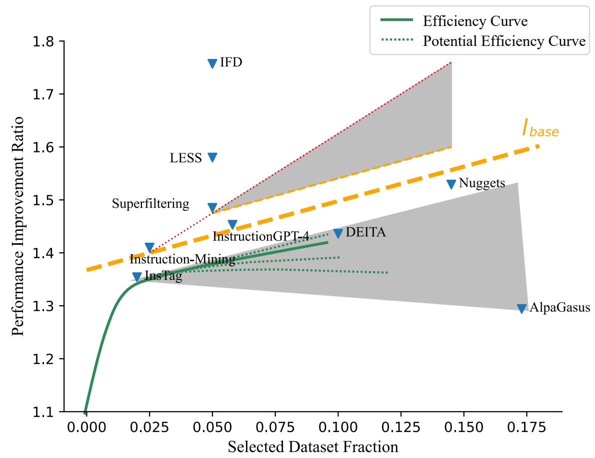

Figure 7: The demonstration of infeasible area. The green line is the efficiency curve of InsTag, where the dashed lines indicate its potential position.

A.3 Infeasible Area

The efficiency curve method generates an infeasible area at each work, which is, in fact, the possible area of its efficiency curve. Therefore, if other work is in the infeasible area, it is incomparable with the work that generates this infeasible area. The infeasible area of the inferior work and superior work is generated differently, where Figure 7 shows respectively by using LESS and InsTag as examples. For both superior and inferior works, the yellow boundary of the infeasible area is parallel to the baseline. For inferior work, the red boundary is horizontal because the SEM should perform at least the same on a larger data size with the same quality. For superior work, the red boundary is the line between the work and Instruction-Mining because the efficiency curve should never penetrate each other (cannot penetrate the baseline here) given the efficiency curve assumption.

All the works are mutually in the feasible area, except DEITA in the infeasible area of InsTag. To determine their relationship of efficiency, we suppose DEITA to be more efficient than InsTag because not only does DEITA have a larger possibility of being better than InsTag according to Figure 7 but also DEITA adopts a more advanced selection method based on InsTag.

A.4 Feasibility

We consider simplicity and transferability as two main aspects when evaluating a selection method's feasibility. This section explains how these two aspects are qualitatively and reasonably evaluated using further refined indicators.

A.4.1 Simplicity

The simplicity of a data selection method takes into account (1) the number of LLMs trained in selector construction, (2) the number of steps in the selection algorithm given a completed selector, and (3) reproducibility, which is based on the open-source state of the code.

# of Trained LLMs. This indicator counts the number of LLMs trained during the selector construction stage. For example, AlpaGasus, InsTag, and Nuggets use purely ChatGPT (commercial LLM) as a scorer or tagger, so the count is 0. IFD, SuperFiltering, and LESS train one warm-up model (LLaMA for IFD and LESS, GPT-2 for SuperFiltering) to obtain quality labels for candidate datasets, so the count is 1. DEITA trains a complexity scorer and a quality scorer from ChatGPT-evolved data separately, so the count is 2. Instruction-Mining fine-tunes 129 models to obtain loss scores on 129 data subsets to rule-fit a linear loss score predictor, so the count is 129. The same count rule applies to InstructionGPT-4 since these two works are almost identical in method.

# of Algorithm Steps. The following pseudo algorithms help count the steps in the selecting stage, where the number in the bracket in the table is the number of LLMs used. For example, based on the Algorithm 1, AlpaGasus performs first ChatGPT scoring and then ranking to get the final selected subset, which consists of 2 steps with 1 LLM usage.

Reproducibility. ✔ means the code is open-source on GitHub, ✘ means the opposite. For example, AlpaGasus has been open-source by others but not by the authors. Thus, we consider it to be close-source. InsTag provides a demo on ModelScope and checkpoints on HuggingFace, but no codes are open-sourced.

A.4.2 Flexibility

The flexibility of a selection method considers both transferability and extensibility. The former corresponds to the question, "Do we need to retrain a selector to maintain optimal performance when changing PFM or Dataset?" while the latter corresponds to "Is the selection method still functional if a certain model is changed?" AlpaGasus's method can use another commercial model other than ChatGPT freely (thus Model Free) and use any dataset they want (thus Dataset Free), but it heavily relies on the existence of at least one commercial model (ChatGPT/GPT-4 as an example). However, for IFD to maintain the ideal selection

Methods	# Trained LLMs	# Algorithm Steps (# Using LLMs)	Rep.	**Simplicity**	Transferability		Extensibility	**Flexibility**	**Feasibility**
					Model Free	Dataset Free	ChatGPT/GPT-4 Free		
AlpaGasus	0	2(1)	✘	**1**	✔	✔	✘	**1**	**1**
InsTag	0	3(1)	✘	**2**	✔	✔	✘	**1**	**2**
Nuggets	0	4(2)	✔	**2**	✘	✔	✔	**2**	**3**
SuperFiltering	1*	3(1*)	✔	**3**	✘	✘	✔	**4**	**4**
IFD	1	3(1)	✔	**4**	✘	✘	✔	**4**	**5**
LESS	1	4(2)	✔	**5**	✘	✘	✔	**4**	**6**
DEITA	2	5(4)	✔	**6**	✔	✘	✘	**3**	**6**
Instruction-Mining	129	4(0)	✘	**8**	✘	✔	✔	**2**	**7**
InstructionGPT-4	30	4(1)	✔	**7**	✘	✔	✘	**5**	**8**

Table 5: Feasibility rank considers both Simplicity rank and Flexibility rank. The former consists of three indicators: (1) # Trained LLMs; (2) # Algorithm Steps (# Times Using LLMs in the algorithm) and (3) Reproducibility, while the latter consider extensibility and transferability. The number in bracket of the "# Algorithm Steps" column indicates the times of LLMs used in the selection algorithm. * indicates that SuperFiltering trains a GPT-2 instead of LLaMA.

performance of the method, one must re-train a selector if either the PFM model or the candidate dataset is changed. Correspondingly, IFD doesn't rely on any commercial model.

Algorithm 1 AlpaGasus

1: **Init** D = Candidate Dataset, S = ChatGPT, U = LLM Usage
2: Use S to score D (U+=1)
–> sample with score
3: Do score ranking and pick top K
4: **Return** Selected Subset

Algorithm 2 InsTag

1: **Init** D = Candidate Dataset, S = ChatGPT, U = LLM Usage
2: Use S to tag D (U+=1)
–> sample with tags
3: Do tag normalization
–> sample with tag statistics
4: Do complexity-first diverse sampling
5: **Return** Selected Subset

Algorithm 3 Nuggets

1: **Init** D = Candidate Dataset, S = PFM, U = LLM Usage
2: Prompt S with zero-shot D (U+=1)
–> sample with ZeroShotScore
3: Prompt S with one-shot D (U+=1)
–> sample with OneShotScore
4: OneShotScore - ZeroShotScore
–> sample with GoldenScore
5: Do score ranking and pick top K
6: **Return** Selected Subset

Algorithm 4 IFD & SuperFiltering

1: **Init** D = Candidate Dataset, S = PFM, U = LLM Usage
2: Use $D' \in D$ to to warm up S
–> pre-experienced S'
3: Use S' to generate IFD/Perplexity score on D (U+=1)
–> each sample with score
4: Do score ranking and pick top K
5: **Return** Selected Subset

Algorithm 5 LESS

1: **Init** D_c = Candidate Dataset, D_t = Target Dataset, S = PFM, U = LLM Usage
2: Use $D'_c \in D_c$ to LoRA warm up S
–> LoRA Model S'
3: Use S' to get gradients of D_c (U+=1)
–> gradient store of D_c
4: Use S' to get gradients of D_t (U+=1)
–> gradient store of D_t
5: Do gradient-similarity-based selection
6: **Return** Selected Subset

Algorithm 6 DEITA

Init D = Candidate Dataset, S = PFM, U = LLM Usage
Use evolved datasets to train two Ss (U+=1)
–> complexity scorer model S_c and quality scorer model S_q
Use S_c to score D (U+=1)
–> instruction with complexity score
Use S_q to score D (U+=1)
–> output with quality score
Multiply two scores and rank
–> ranked sample
Do score-first, diversity-aware selection (U+=1)
Return Selected Subset

Algorithm 7 Instruction-Mining

Init D_c = Candidate Dataset, D_t = Training Dataset, S = Linear Selector, U = LLM Usage

Use vectorized D_t to train a linear selector
–> trained S
Do vectorization on D_c with indicators
–> vectorized D_v
Use S to predict loss on D_v
–> sample with loss score
Do score ranking and pick top K
Return Selected Subset

Algorithm 8 InstructionGPT-4

Init D_c = Candidate Dataset, D_t = Training Dataset, S = Transformer model, U = LLM Usage
Use vectorized D_t to train a self-attention NN (U+=1)
–> trained S
Do vectorization on D_c with indicators
–> vectorized D_v
Use S to predict loss on D_v
–> sample with loss score
Do score ranking and pick top K
Return Selected Subset