DataCurBench: Are LLMs Ready to Self-Curate Pretraining Data?

Anonymous ACL submission

Abstract

The quality of pre-training corpora is central to the capabilities of large language models (LLMs), yet current curation pipelines that rely on rule-based filters or small supervised models lack scalability and adaptability. This work introduces **DataCurBench**, a comprehensive benchmark for evaluating the ability of LLMs to autonomously perform two sequential pre-training data curation tasks: **data filtering**, which selects high-quality training data, and **data cleaning**, which improves linguistic form and coherence to enhance training effectiveness. We propose a systematic evaluation framework and present empirical findings that reveal a dual pattern in LLM performance. While LLMs demonstrate near-human proficiency in language-driven data cleaning, they remain limited in data filtering, often failing to consistently apply prompt-based selection criteria and underperforming compared to fine-tuned smaller models. DataCurBench is publicly available[1], offering a practical benchmark to evaluate data curation, highlight key challenges, and support the development of more efficient and ethical pre-training pipelines.

1 Introduction

Large language models (LLMs) have driven breakthroughs in NLP tasks such as text generation, machine translation, and complex question answering. However, their outstanding performance depends on the quality of training data. Training corpora, sourced from diverse datasets like CommonCrawl, Wikipedia, arXiv, and GitHub, are rich but contain challenges including redundancy, harmful or sensitive content, and cross-domain inconsistencies. Therefore, robust and ethical preprocessing pipelines are essential to ensure data representativeness and adherence to quality and ethical standards (Radford et al., 2019; Brown et al., 2020).

Traditional data curation uses **a sequential pipeline of data filtering followed by data cleaning** to tackle scale, quality, and ethical challenges. After initial preprocessing like URL deduplication and language identification, raw corpora first undergo filtering based on heuristics (e.g., perplexity thresholds) (Gao et al., 2020) or small fine-tuned models trained on curated data with raw web negatives (Radford et al., 2019). Recent advances incorporate machine learning for large-scale content filtering, including toxicity detection, and use structured annotation frameworks like Redpajama (Gehman et al., 2020) and Fineweb-edu (Lozhkov et al., 2024) to improve classifier training and evaluation. Data cleaning then applies sequence-to-sequence approaches to normalization, grammar, style, and factual consistency. Nevertheless, these pipelines still face challenges in effectiveness and scalability as corpora grow beyond billions of documents (Yuan et al., 2023).

Against this backdrop, a compelling research question emerges: *Can LLMs autonomously curate their training data?* If so, this could revolutionize preprocessing by reducing human effort and adapting to new data distributions. Rigorous validation requires a systematic evaluation framework covering key data curation tasks.

To this end, we introduce DataCurBench, a comprehensive benchmark specifically designed to assess the self-curation capabilities of LLMs. Unlike previous benchmarks that predominantly target rule-based or small-model preprocessing techniques, DataCurBench establishes a structured framework for evaluating LLMs in tasks such as harmful content identification, redundancy reduction, and the enhancement of data diversity and quality. Our benchmark further incorporates novel metrics that quantify the alignment between LLM-curated datasets and traditional human-curated standards, thereby providing a precise measure of both efficacy and ethical compliance.

[1] https://huggingface.co/datasets/anonymousaiauthor/DataCurBench

Preliminary results show that LLMs vary in their ability to handle data curation tasks with the same objectives but different formats. They **excel at data cleaning tasks** via text rewriting, likely owing to their strong language modeling, contextual reasoning, and alignment with instruction-tuned generation objectives. In contrast, their **filtering performance**, posed as scoring or classification, **remains limited**, perhaps because prompt criteria lack the precision or clarity for LLMs to grasp filtering rules without explicit fine-tuning. These findings underscore LLMs' potential to cut manual preprocessing substantially and identify where additional fine-tuning or supervision is needed.

2 Dataset Construction

For comprehensive evaluation, DataCurBench comprises two tasks, filtering and cleaning, and draws on two data sources: manually annotated samples from open-source datasets and LLM-generated synthetic samples covering diverse scenarios.

2.1 Real-world Sample Annotation

Real-world samples are drawn from two widely used corpora: the English RedPajama-Data-V2 dataset (Weber et al., 2024) and the Chinese CCI3-Data corpus (Wang et al., 2024), chosen for their large-scale pretraining use, rich metadata, and realistic distributions. We first visualized key metadata (e.g., perplexity, heuristic scores) to characterize each corpus, then applied stratified sampling to select **1,000** samples per dataset that preserve the original distributions.

The annotation followed a unified framework based on four criteria, Sensitive Information Safety (**SI**), Content Clarity and Integrity (**CC**), Formatting Consistency (**FC**), and Content Relevance (**CR**), drawn from established pretraining quality control practices and insights from iterative data processing and manual review workflows in large model training (e.g., LLaMA (Touvron et al., 2023), Qwen (Yang et al., 2024), Yi (Young et al., 2024)).

Each sample was independently scored (0–5) on each criterion by **at least two annotators** and assigned a binary *retain*/*reject* label reflecting pretraining utility; disagreements were resolved by a third expert or replaced by a same-stratum sample if consensus failed. This yielded **733/267** retained/rejected Chinese samples and **596/404** English samples for the filtering task. Detailed criteria definitions, guidelines, and example annotated data samples appear in Appendix A.

2.2 Synthetic Sample Generation

We initially applied manual annotation for the cleaning task on real-world datasets using a similar protocol but found it impractical. First, many "rejected" samples were too **noisy** and typically discarded before cleaning, making them **unsuitable for cleaning evaluation**; second, even with detailed criteria, annotations for cleaning **lacked consistency** and were **subjective**; third, sampled datasets provided **insufficient coverage** of real-world issues. These limitations prompted a shift toward synthetic sample generation.

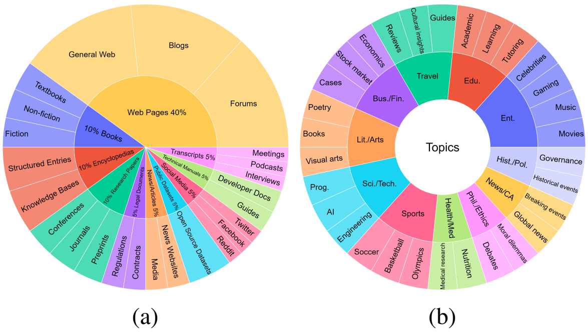

Figure 1: Distribution of data sources (a) and topic hierarchy (b) in the synthetic dataset. The proportions of data sources are specified, while sub-sources, topics, and subtopics follow a uniform distribution.

Considering the unique challenges of cleaning tasks and the nature of real-world data, our dataset design follows three principles: diversity of data sources, breadth of topic coverage, and realism in preprocessing challenges. Building on structured prompt methodologies from recent synthetic dataset work (Gunasekar et al., 2023), we use a single template with multiple placeholders, each tied to a semantic content property or corruption type. These placeholders are systematically instantiated from predefined sets to produce a diverse and realistic collection of generation prompts.

Semantic content properties are ederived by analyzing the composition of representative LLM corpora, as documented in prior literature (Touvron et al., 2023; Yang et al., 2024; Young et al., 2024), yielding ten source types (e.g., web, books) and eleven domains (e.g., Science, Health) with estimated proportions (e.g., 40% web, 10% books). These are refined into fine-grained subtypes via GPT-4o, producing the source distribution (Figure 1a) and topic taxonomy (Figure 1b), ensuring semantic specificity and domain relevance in prompt construction.

Beyond semantic diversity, corruption patterns are modeled from open-source datasets and industrial toolkits (Weber et al., 2024; Chen et al., 2024), yielding 30 data processing functions grouped into basic, moderate, and advanced levels. These cover tasks from simple sanitization to complex normalization. We also define two key challenge types, *Character Corruption* and *Edge Cases*, to capture issues under-represented in existing resources. These challenges are randomly injected: 20% each for the two types, 10% both, and 50% left clean.

Using GPT-4o with diverse prompts, we synthesized **3,000** English and Chinese samples; the prompt templates and definitions of processing functions, and challenges are in Appendix B.

3 Evaluation Pipeline

Grounded in our dataset design—real annotations for filtering and synthetic data for cleaning—we develop a unified evaluation pipeline with two parallel tracks. Standardized inputs, prompts, and criteria ensure fair and reproducible comparisons across LLMs and traditional baselines.

Specifically, our evaluation follows four integrated steps. First, we design task-specific prompts that encode the four quality metrics (**SI**, **CC**, **FC**, **CR**) and provide illustrative output examples. Second, these prompts are applied to LLMs using best-practice hyperparameters, while traditional baselines follow their standard protocols (e.g., heuristics or compact models trained on high-quality corpora), with no task-specific fine-tuning. Third, output formats are constrained: models generate binary retain/reject labels for filtering and rewritten texts for cleaning, enabling deterministic parsing. Finally, filtering is assessed via **absolute classification metrics** against human-annotated ground truth, while cleaning quality is evaluated via pairwise comparisons against reference rewrites from an LLM, judged by another LLM across the four metrics, yielding **dimension-wise** and **overall win rates** as **relative performance** indicators.

3.1 Data Filtering Track

The data filtering track evaluates a model's ability to distinguish between text samples suitable for pretraining and those that should be discarded, aligning with practical requirements in large-scale data curation. The task is formulated as a binary classification problem: given an input sample, the model must decide to *Retain* or *Reject* it based on the four predefined quality criteria.

Notably, as specified in the annotation guidelines, the goal is not perfection. To reflect the practical realities of curating pretraining corpora, the label *Retain* is assigned to samples that demonstrate a clear **educational or informational value** and meet an acceptable quality threshold, despite minor formatting or clarity flaws. Conversely, the *Reject* label applies to samples with either **serious flaws** in any single criterion (e.g., harmful content, incoherence) or an **overall lack of value** (e.g., irrelevant or meaningless text).

Models are evaluated under uniform inference settings without task-specific tuning. A structured prompt (Appendix C.1, Figure 9) encodes the four annotation criteria, clarifies tolerance for minor flaws if educational value is present, and constrains output to a single `<answer>Retain/Reject</answer>` tag. This ensures deterministic parsing and aligns model decisions with human judgment while isolating model capability from implementation factors.

Given the importance of filtering low-quality or harmful content, evaluation focuses on **Precision**, **Recall**, and **F1** for **rejected samples**, using human annotations as ground truth. These metrics reflect the model's ability to correctly identify and exclude irrelevant, incoherent, or sensitive data, aligning with real-world needs in large-scale pretraining corpus construction.

3.2 Data Cleaning Track

The data cleaning track evaluates a model's capability to enhance text quality through addressing grammar, clarity, and coherence while maintaining semantic fidelity. The task is formulated as a **text rewriting** problem, requiring models to produce outputs that **satisfy all four predefined quality metrics** beyond simple formatting corrections.

In practical data cleaning workflows, downstream model performance on cleaned data serves as the most reliable quality indicator, making absolute scoring insufficient. Therefore, reference-free evaluation metrics are essential. AlpacaEval (Li et al., 2023) is an automated, reference-free method that applies fixed instructions to both a strong baseline and the evaluated model, then uses a separate LLM judge to estimate the preference probability. The resulting win rate reflects relative model performance. Inspired by this approach, we propose a systematic comparative framework for evaluating data cleaning on our synthetic dataset.

To ensure uniform evaluation, we adopt a structured prompt (Appendix C.2, Figure 10) mirroring the filtering track. This prompt incorporates precise definitions of the four quality criteria, reframed as rewriting objectives rather than classification rules, and mandates output wrapped in `<clean_text>. . . </clean_text>` tags for deterministic parsing.

Using this setup, we prompt both candidate and reference models to generate cleaned outputs from identical inputs under uniform, non-finetuned inference settings, isolating inherent cleaning capability differences. We select **GPT-4** as the reference model due to its consistent **near-human output quality** and favorable **cost-performance balance**, providing a stable and practical baseline without complex comparison schemes.

Outputs are evaluated through pairwise comparisons by a dedicated judge model, which selects the superior rewrite between the evaluated and reference models given the same input. We use **GPT-4o** as judge due to its strong reward modeling capability and over **90% agreement with human preferences** on a sampled subset. Importantly, the judge is distinct from the reference model to **reduce bias** and ensure evaluation reliability.

The judgment follows a structured evaluation prompt (Appendix C.3, Figure 11), which also retains the definitions of the four quality criteria, while reframing them as comparative decision rules. To mitigate positional bias, the presentation order of candidate and reference outputs is randomized. Results are reported as **win rates per dimension** (SI, CC, FC, CR) and **overall**, reflecting relative cleaning performance.

4 Results

We evaluate the native capabilities of unfine-tuned, open-source LLMs under 10 billion parameters on data filtering and cleaning tracks, addressing practical challenges in large-scale, bilingual corpus curation. Our selection includes seven transparently developed model series supporting English and Chinese, *LLaMA-3, Mistral, Phi, Gemma-2, Qwen2.5, MiniCPM, and Yi-1.5*, each tested with base and supervised fine-tuned (SFT) variants. Leveraging strong few-shot adaptation, all models received limited examples and precise instructions per track, without parameter updates. Model names, API versions, and HuggingFace organizations used for reproducibility are listed in Appendix D.1, Table 5.

Under few-shot prompting, sub-10 billion-parameter LLMs show moderate performance in both filtering and cleaning tasks, indicating significant scope for improvement. These models establish a baseline for large-scale data curation, yet further tuning and specialized methods are required to fulfill practical pretraining demands.

Figure 2 shows precision–recall curves for all seven model series on the data filtering track, while Figure 3 presents their overall win rates on the data cleaning track relative to GPT-4, evaluated by GPT-4o. Results are compared against human performance, traditional pipelines, and API baselines, highlighting relative model capabilities.

On the filtering track, precision–recall contours reveal a clear performance hierarchy, where equal-F1 isolines indicate comparable overall performance and higher recall reflects a more favorable trade-off. Leading open-source models such as Meta-Llama-3-8B-Instruct and Qwen2.5-7B-Instruct modestly outperform most peers but remain notably behind traditional pipelines and human annotations. Closed-source APIs (GPT-4o, Gemini, and Claude 3.5) widely regarded as stronger exceed open-source models but still perform comparably to traditional methods and below human levels. These findings highlight that, given current accuracy and computational constraints, neither open- nor closed-source LLMs yet surpass established pipelines or human judgment for practical filtering applications.

In contrast, on the data cleaning track, open-source models exhibit markedly stronger performance. Models such as gemma-2-9B-it and Qwen2.5-7B-Instruct achieve results comparable to human annotators and closed-source APIs, significantly closing the gap. Notably, a broad range of models, including many with only 3B or even smaller parameters, also surpass traditional pipelines. These results indicate that, in terms of both effectiveness and cost, LLM-based data cleaning, especially with small and mid-sized open models, offers a practical and scalable alternative to traditional pipelines and manual efforts.

We further investigated the detailed patterns behind these findings by analyzing the influence of key model characteristics such as instruction tuning, model scale, architectural design, and multilingual training. This analysis demonstrates how these factors influence performance in data filtering and cleaning tasks.

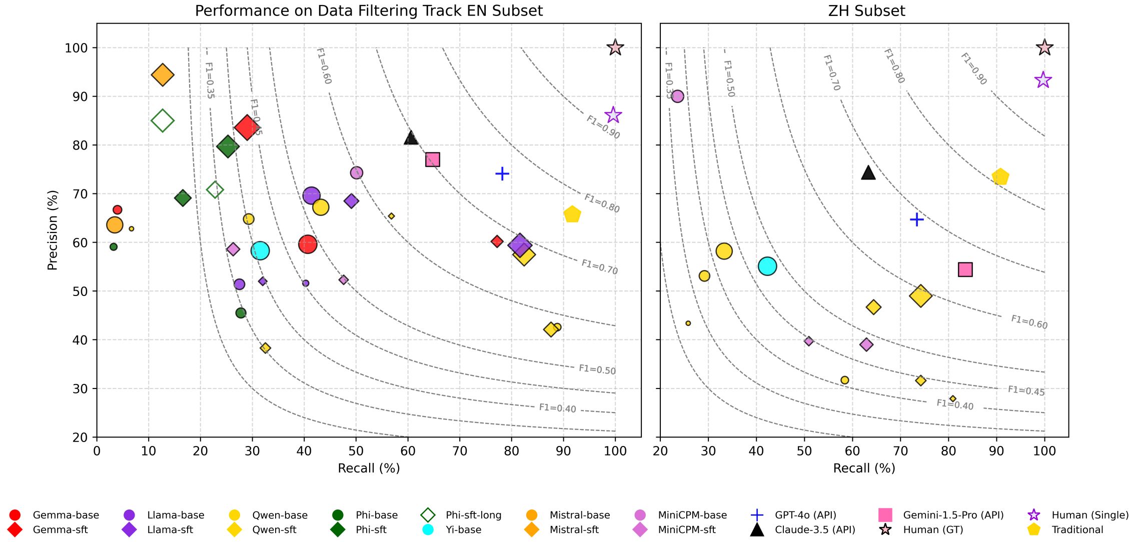

Figure 2: Few-shot Precision-Recall on Data Filtering track, across model series (color), model types (shape: **circle** for base models, **diamond** for SFT models, **star** for Human, **pentagon** for Traditional and **other markers** for APIs), and model sizes (marker size). Baselines: Human(single) - average performance of two annotators; Human (GT) - ground truth; Traditional - separate educational quality classifiers for Chinese and English, trained on annotations generated by LLaMA3-70B-Instruct and Qwen-72B-Instruct, according to (Lozhkov et al., 2024)

Impact of Instruction Tuning: Instruction-tuned models exhibit notable improvements in both data filtering and data cleaning tracks. In the filtering track (Figure 2), instruction tuning primarily boosts recall while preserving precision at parity, yielding consistent F1 improvements across model series (LLaMA-3, Gemma-2, Phi, and Qwen2.5) in both English and Chinese subsets. The effect is even more striking in the cleaning track (Figure 3), where instruction-tuned models consistently surpass their base version at every scale and across all families, as evidenced by the upward shift of vertically paired points. These gains transcend language boundaries and even enable smaller SFT models to outperform larger base models. In particular, on the cleaning task, SFT models not only more frequently outperform traditional pipeline-based approaches but also approach human-level quality. Together, these findings underscore the critical role of instruction tuning in aligning model outputs with human judgments of data quality.

Impact of Model Size: The expected positive correlation between model size and performance emerges more consistently in the data cleaning track than in the data filtering track. In the cleaning setting, larger models (represented by larger markers) consistently outperform their smaller counterparts within the same model series (color) and type (shape). This hierarchy persists across settings, as larger models often surpass smaller models even when the latter benefit from otherwise favorable configurations. In contrast, the filtering track shows more deviations: within the Gemma-2 and Qwen2.5 series, smaller models frequently match or exceed the performance of mid to large scale variants, particularly on the English subset. When other configurations are held constant, the overall performance gap between smaller and larger models in filtering remains minimal, obscuring any clear size effect. These observations imply that model size plays a more decisive role in data cleaning performance, whereas its influence on filtering is moderated by factors such as prompt sensitivity and overfitting to spurious patterns.

Impact of Model Series: Model series exert a pronounced influence on performance across both tracks. In the filtering track, the LLaMA-3 and Qwen2.5 series exhibit consistently high recall and precision, matching or exceeding other leading series. In the cleaning track, series-level differences are even more striking: the Gemma-2 series dominates the English subset, while the Qwen2.5 series leads in the Chinese subset. Other families, such as Phi and Yi-1.5, remain competitive but generally underperform relative to the top series, even when controlling for model size and instruction tuning. These observations underscore the substantial role of model family characteristics, including architectural design and pretraining data composition, in determining performance on both data filtering and

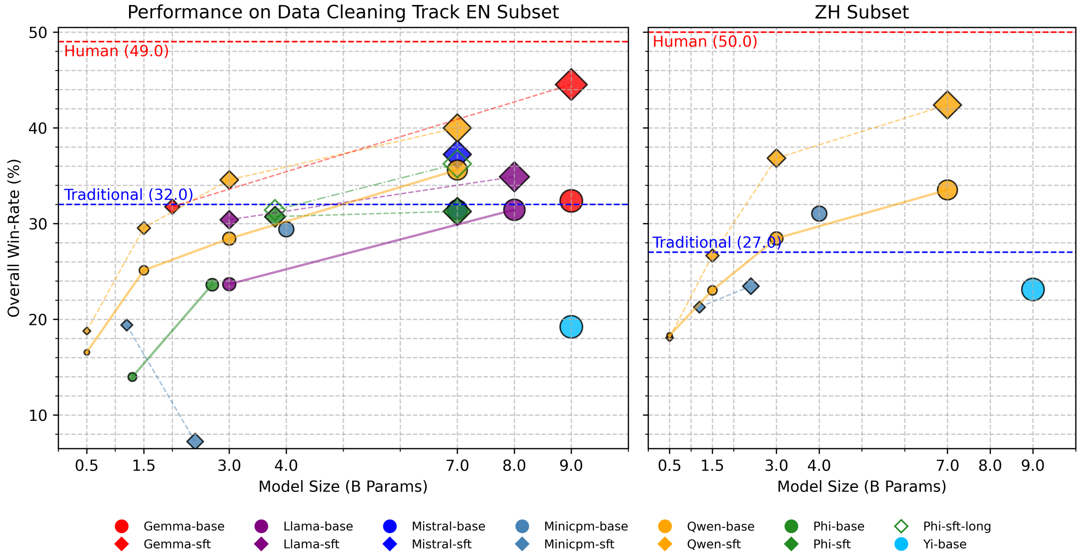

Figure 3: Few-shot Overall Win Rates on Data Cleaning track (relative to GPT-4) judged by GPT-4o, across model series (color), model types (shape: **circle** for base models, **diamond** for SFT models, **star** for Human, and **pentagon** for Traditional), and model sizes (marker size). Baselines: Human - human-annotated clean texts; Traditional - results produced by the default Data-Juicer (Chen et al., 2024) cleaning configuration, a widely adopted, representative LLM data-processing system covering diverse cleaning functions.

cleaning tasks.

Impact of Language: Evaluation language exerts minimal influence on the relative performance ranking of bilingual models across both tasks. Within each series, precision and recall contours for English and Chinese filtering tasks closely coincide, and model orderings in the cleaning track remain largely stable across languages. For instance, Qwen-2.5-7B-Instruct occupies the top position in both English and Chinese subsets, achieving nearly identical filtering F1 scores and cleaning win rates. These findings indicate that balanced multilingual pretraining and prompt design mitigate language-specific variations in model behavior.

4.1 Performance Across Evaluation Metrics

As illustrated in Figure 2, instruction-tuned models achieve a more balanced precision–recall trade-off, whereas base models remain overly conservative, which prioritize precision at the expense of drastically reduced recall. Closed-source APIs, such as GPT-4o, deliver modestly higher recall while sustaining moderate precision, indicative of stronger instruction adherence. Nevertheless, overall effectiveness remains constrained: most models exhibit recall **below 50%**, and only **a few exceed 80%**. Since the primary objective of data filtering is to excise harmful content without discarding valuable data, recall is of central importance, even at the cost of imperfect precision. The widespread low-recall tendency thus reveals a fundamental shortcoming in current model strategies for the filtering task.

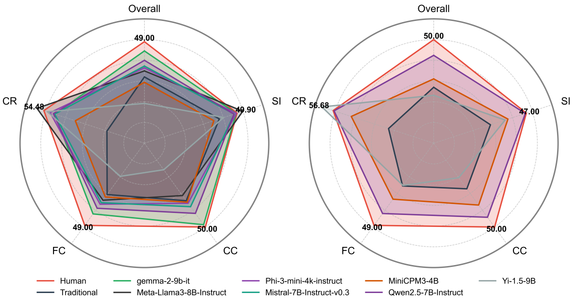

Figure 4: Cleaning performance radars across metrics on EN (left) and ZH (right) subset.

For the cleaning track, we selected each series' top-performing model and plotted their win rates across four quality dimensions, Sensitive Information Safety (SI), Content Relevance (CR), Content Clarity and Integrity (CC), and Formatting Consistency (FC), as well as an overall score (Figure 4). The models demonstrate pronounced strengths in SI and CR, achieving win rates above 50 % and even surpassing human annotations and the reference model, which likely reflects pretrained models' ability to preserve meaning through precise, minimal edits. In contrast, although performance on CC and FC exceeds that of traditional rule-based systems, it remains below human-level quality. These findings underscore current models' limitations in ensuring linguistic clarity and formatting consistency while highlighting their potential to outperform humans in maintaining core semantic content. Full win rate statistics are provided in Appendix D.2 Table 6.

4.2 Effect of Few-Shot Prompting

We evaluated the impact of in-context examples on both tracks across several model families (Table 1; additional results in Appendix D.3 Table 7). The benefits of six-shot prompting decrease as model capacity and instruction tuning increase. For instance, LLaMA-3.2-1B-Instruct's cleaning win rate rises from **14.5%** to **19.9%** and its filtering performance from **12.0%** to **39.6%**, corresponding to improvements of **5.4** and **27.6** percentage points, respectively. In contrast, Qwen-2.5-7B-Instruct attains **45.3%** (cleaning) and **61.5%** (filtering) under zero-shot conditions, with marginal changes when supplied with examples (**–5.3** points for cleaning; **+6.3** points for filtering). Base models derive substantially greater gains from prompting: Qwen-2.5-3B's filtering accuracy jumps from **6.6%** to **60.4%** with six examples, a **53.8**-point increase, whereas its instruction-tuned counterpart improves by only **3.2** points.

These trends persist in both English and Chinese subsets, indicating that few-shot examples are particularly effective for smaller or untuned models in enhancing data processing capabilities. Considering the massive pretraining data scale and computational costs, few-shot prompting offers a practical means to enable LLMs to perform more effective data curation without full-scale retraining.

4.3 Cleaning Performance by Metadata

To better understand model capabilities in data cleaning, we partitioned the benchmark into subsets defined by metadata from the dataset construction process. These dimensions include **data source** and **topic**, capturing stylistic and semantic properties, as well as **difficulty** and **challenge type**, reflecting noise severity and nature. Performance across these subsets highlights the impact of data characteristics on model robustness.

Figure 5 presents a heatmap of model performance by source and topic. Inputs drawn from structured sources, research papers, encyclopedias, and technical manuals, consistently achieve higher scores, indicating strong alignment with models pretrained on formal text. In contrast, unstructured inputs such as social media posts and public datasets yield lower performance, underscoring the difficulty of processing informal and noisy text. By topic, models excel in abstract domains (literature, arts, philosophy) but struggle in specialized fields (sports, business, mathematics). Particularly low scores in health, medicine, and science suggest that handling technical content effectively requires targeted fine-tuning.

Table 1: Performance of 6-shot vs. 0-shot on Cleaning and Filtering; **bold**: **unexpected 0-shot wins**

Model	Cleaning		Filtering	
	6-shot	0-shot	6-shot	0-shot
EN Subset				
gemma-2-9b-it	44.53	41.58	43.09	29.94
gemma-2-9b	32.37	**34.77**	48.38	10.26
gemma-2-2b-it	31.77	**33.53**	67.61	47.57
gemma-2-2b	26.57	25.58	7.49	**28.76**
Meta-Llama-3-8B-Instruct	34.90	**40.48**	68.76	41.40
Meta-Llama-3-8B	31.45	30.65	51.94	47.22
Llama-3.2-3B-Instruct	30.40	20.62	57.23	9.86
Llama-3.2-3B	23.67	23.38	35.86	4.82
Llama-3.2-1B-Instruct	19.90	14.52	39.63	12.98
Llama-3.2-1B	18.35	7.97	45.27	28.70
Qwen2.5-7B-Instruct	40.00	**45.25**	67.76	61.50
Qwen2.5-7B	35.60	**36.48**	52.57	31.47
Qwen2.5-3B-Instruct	34.57	30.93	60.82	**63.94**
Qwen2.5-3B	28.43	**28.55**	40.34	6.64
Qwen2.5-1.5B-Instruct	29.55	21.18	56.84	52.78
Qwen2.5-1.5B	25.12	20.92	57.56	7.55
Qwen2.5-0.5B-Instruct	18.78	13.37	35.17	20.73
Qwen2.5-0.5B	16.55	14.60	12.11	0.50
ZH Subset				
Qwen2.5-7B-Instruct	42.40	**45.13**	59.02	**60.53**
Qwen2.5-7B	33.52	33.52	42.38	13.70
Qwen2.5-3B-Instruct	36.83	36.58	54.17	**60.40**
Qwen2.5-3B	28.45	**28.60**	37.68	12.46
Qwen2.5-1.5B-Instruct	26.67	25.87	44.34	**49.35**
Qwen2.5-1.5B	23.03	**23.15**	41.11	40.85
Qwen2.5-0.5B-Instruct	18.38	15.57	41.54	26.16
Qwen2.5-0.5B	18.32	16.35	32.39	7.02

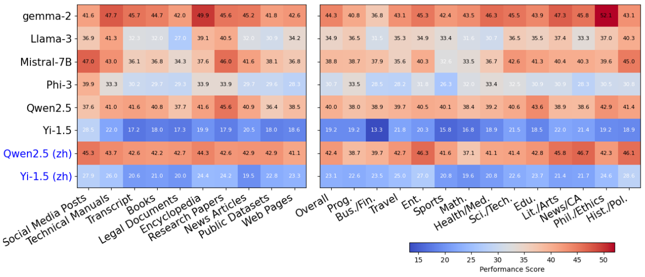

Figure 5: Cleaning performance heatmaps across data sources (left) and topics (right).

We then analyze cleaning performance by task difficulty (Table 2) and by challenge type in noise (Table 3). In Table 2, every model exceeds its

Table 2: Performance across difficulty levels: Basic, Moderate, Advanced; ↑ / ↓: above / below average score

Model	Basic	Moderate	Advanced
EN Subset			
gemma-2-9b-it	48.88$_{\uparrow}$	42.10$_{\downarrow}$	39.67$_{\downarrow}$
Meta-Llama-3-8B-Instruct	36.64$_{\uparrow}$	29.25$_{\downarrow}$	30.67$_{\downarrow}$
Mistral-7B-Instruct-v0.3	40.80$_{\uparrow}$	39.10$_{\downarrow}$	35.00$_{\downarrow}$
Phi-3-mini-4k-instruct	33.88$_{\uparrow}$	30.20$_{\downarrow}$	26.20$_{\downarrow}$
Qwen2.5-7B-Instruct	43.40$_{\uparrow}$	38.65$_{\downarrow}$	36.13$_{\downarrow}$
Yi-1.5-9B	22.56$_{\uparrow}$	15.85$_{\downarrow}$	18.13$_{\downarrow}$
ZH Subset			
Qwen2.5-7B-Instruct	44.24$_{\uparrow}$	40.65$_{\downarrow}$	41.67$_{\downarrow}$
Yi-1.5-9B	24.88$_{\uparrow}$	22.15$_{\downarrow}$	21.53$_{\downarrow}$

Table 3: Performance across challenge scenarios, CC for Character Corruption challenges and EC for Edge Case challenges, w/o means no challenge injected , ↑ / ↓: above / below average score

Model	w/o	CC	EC	CC&EC
EN Subset				
gemma-2-9b-it	45.85$_{\uparrow}$	41.27$_{\downarrow}$	47.25$_{\uparrow}$	37.15$_{\downarrow}$
Meta-Llama-3-8B-Instruct	32.60$_{\downarrow}$	32.84$_{\downarrow}$	43.53$_{\uparrow}$	33.55$_{\downarrow}$
Mistral-7B-Instruct-v0.3	38.14$_{\downarrow}$	36.72$_{\downarrow}$	44.21$_{\uparrow}$	35.52$_{\downarrow}$
Phi-3-mini-4k-instruct	37.00$_{\uparrow}$	32.34$_{\downarrow}$	43.62$_{\uparrow}$	27.17$_{\downarrow}$
Qwen2.5-7B-Instruct	39.67$_{\downarrow}$	37.14$_{\downarrow}$	46.66$_{\uparrow}$	34.37$_{\downarrow}$
Yi-1.5-9B	19.39$_{\uparrow}$	15.63$_{\downarrow}$	24.85$_{\uparrow}$	14.57$_{\downarrow}$
ZH Subset				
Qwen2.5-7B-Instruct	38.38$_{\downarrow}$	46.75$_{\uparrow}$	36.01$_{\downarrow}$	43.61$_{\uparrow}$
Yi-1.5-9B	24.59$_{\uparrow}$	17.04$_{\downarrow}$	28.49$_{\uparrow}$	17.68$_{\downarrow}$

overall mean on basic tasks (for example, Gemma-2-9B-Instruct achieves **48.9%**), yet accuracy declines steadily on moderate tasks (down to **42.1%**) and advanced tasks (as low as **39.7%**). Table 3 shows that isolated edge-case challenges (**EC**) often match or surpass the original baseline (Gemma-2-9B-Instruct: **47.3%** versus **45.9%**), whereas character corruption alone (**CC**) reduces scores by **4** to **6 percentage points** and the combined edge-case plus character corruption (**EC+CC**) scenario experiences the largest drops (e.g., Phi-3-Mini at **27.2%**). These findings indicate that although current LLMs can effectively clean simple or isolated noise, they remain highly sensitive to increased task complexity and overlapping structural and contextual distortions, pointing to critical directions for improving robustness.

5 Related Work

Pre-training data processing for Large Language Models (LLMs) is crucial for improving model performance. As LLMs are trained on massive datasets, ensuring high-quality data is essential to prevent issues such as bias and irrelevant content. (Brown et al., 2020) emphasized the importance of filtering low-quality data from web-crawled datasets to mitigate risks such as offensive language and bias . Similarly, (Queiroz Abonizio and Barbon Junior, 2020) introduced a technique using GANs to reduce noise in text data, which can improve model performance.

The automation of data preprocessing has also been a significant focus. (Ouyang et al., 2022) proposed an unsupervised data filtering framework using clustering techniques to identify high-quality data points (Ouyang et al., 2022). (Gupta and Mahmood, 2024) further extended this concept by developing data augmentation methods to diversify training data , thus improving model robustness.

Automated data annotation has also been explored. (Gururangan et al., 2020) showed how pre-trained LLMs can be used to automatically label large datasets, reducing the need for manual labeling. Additionally, (Celikyilmaz et al., 2020) introduced a Data Quality Score (DQS) metric to assess text quality, which allows for standardized evaluation of preprocessing methods.

Evaluation of data preprocessing techniques is critical for understanding their effectiveness. (Elangovan et al., 2024) developed a framework that combines human and machine evaluations of model performance after training on processed datasets . Comparative studies, such as (BARBERIO, 2022), have shown that tailored data cleaning techniques aligned with specific tasks result in better downstream performance .

These advancements in data processing and evaluation contribute to the development of more robust and efficient LLMs, particularly by focusing on automation, quality control, and comprehensive evaluation metrics.

6 Conclusion

This study introduces DataCurBench for evaluating the autonomous curation capabilities of LLMs on pretraining data. We find that LLMs deliver near-human quality in data cleaning owing to their strong language comprehension and instruction-following skills, yet they struggle in data filtering because they cannot align selection criteria with human preferences specified in prompts. These results highlight the complexity of data filtering and the need for fine-tuning or minimal supervision to close this performance gap, while demonstrating that LLMs can significantly reduce manual and rule-based preprocessing and enable automated iterative improvements in pretraining workflows.

7 Limitations

DataCurBench has two tracks: data filtering and data cleaning. In the data filtering track, expert annotators provided ground truth labels via independent blind annotation and consensus adjudication, though individual differences in judgment criteria and domain expertise may still introduce subjective bias into the labels. In the data cleaning track, GPT-4o served as the sole judge model, achieving over 90 percent agreement with human evaluators, yet reliance on a single large language model may embed biases specific to that model, such as over sensitivity to certain linguistic patterns or artifacts of its training data, into the assessment. The synthetic and sampled corpora used in both tracks cannot fully represent the heterogeneity and complexity of real world datasets. Future work will extend evaluation to diverse large scale real world corpora and employ a heterogeneous ensemble of human experts and multiple automated evaluators to mitigate bias, improve fairness and more rigorously assess the generalization of LLM driven curation.

DataCurBench are released under Apache 2.0, but original source datasets (CCI3.0-Data[2], RedPajama-Data-V2[3]) retain their original licenses; users must honor those terms. All datasets were used according to their intended research purposes, and our derivatives are intended for academic use; commercial or other uses require separate license checks.

We initially applied automated PII detection (e.g., emails, phone numbers) and offensive-language filters, complemented by manual review, to our original samples; however, some residual sensitive or harmful content may persist. Additionally, our synthetic datasets intentionally include instances of “fake" PII and harmful content to rigorously evaluate filtering and cleaning pipelines. Users should exercise caution and perform any further required anonymization and validation when using these data.

[2] https://huggingface.co/datasets/BAAI/CCI3-Data

[3] https://huggingface.co/datasets/togethercomputer/RedPajama-Data-V2

References

ANNA BARBERIO. 2022. Large language models in data preparation: opportunities and challenges.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. volume 33, pages 1877–1901.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

D. Chen, Y. Huang, Z. Ma, et al. 2024. Data-juicer: A one-stop data processing system for large language models. In *Companion of the 2024 International Conference on Management of Data*, pages 120–134.

Aparna Elangovan, Ling Liu, Lei Xu, Sravan Bodapati, and Dan Roth. 2024. Considers-the-human evaluation framework: Rethinking human evaluation for generative large language models. *arXiv preprint arXiv:2405.18638*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

S. Gunasekar, Y. Zhang, J. Aneja, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Parul Gupta and Maha Mahmood. 2024. Data augmentation for natural language processing. *International Journal of Data Science and Advanced Analytics*, 6(6):352–359.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don’t stop pretraining: Adapt language models to domains and tasks.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

Xu Ouyang, Shahina Mohd Azam Ansari, Felix Xiaozhu Lin, and Yangfeng Ji. 2022. Efficient model finetuning for text classification via data filtering. *arXiv preprint arXiv:2207.14386*.

Hugo Queiroz Abonizio and Sylvio Barbon Junior. 2020. Pre-trained data augmentation for text classification. In *Brazilian Conference on Intelligent Systems*, pages 551–565. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hugo Touvron, Thibault Lavril, Gaëtan Izacard, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Liangdong Wang, Bo-Wen Zhang, Chengwei Wu, Hanyu Zhao, Xiaofeng Shi, Shuhao Gu, Jijie Li, Quanyue Ma, TengFei Pan, and Guang Liu. 2024. Cci3. 0-hq: a large-scale chinese dataset of high quality designed for pre-training large language models.

Maurice Weber, Daniel Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. *Preprint*, arXiv:2411.12372.

A. Yang, B. Yang, B. Zhang, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

A. Young, B. Chen, C. Li, et al. 2024. Yi: Open foundation models by 01.ai. *arXiv preprint arXiv:2403.04652*.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, Fangyuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations. *Advances in Neural Information Processing Systems*, 36:58478–58507.

A Details in Real-world Sample Annotation

To ensure consistent and high-quality annotations, each text sample was assessed according to the following four criteria. Each criterion is scored on a scale from 0 to 5, where 0 represents a severe deficiency and 5 reflects optimal quality. The scores are intended to capture various levels of quality across dimensions and guide the final decision to either *Retain* or *Reject* the sample.

A.1 Evaluation Criteria

Sensitive Information Safety(SI): This criterion measures the presence of sensitive or harmful content, such as personal information, offensive language, or other material that could pose a privacy risk. Higher scores correspond to the absence of such content, ensuring that the text is safe for use in training models.

- **0:** Contains significant sensitive or harmful content, such as identifiable personal data or hate speech.
- **1-2:** Contains noticeable sensitive content that may hinder usability or introduce ethical concerns.
- **3-4:** Minor sensitive content present, but overall suitable for training.
- **5:** No identifiable sensitive or harmful content.

Content Clarity and Integrity (CC): Evaluates whether the core meaning and key information of the text are preserved, ensuring that the text is free from significant ambiguity, factual distortion, or loss of meaning.

- **0:** Severe loss of meaning or major factual distortions.
- **1-2:** Major ambiguities or distortions affecting clarity and usefulness.
- **3-4:** Minor issues, but the core meaning is largely preserved.
- **5:** Fully clear and accurate, with no ambiguity or distortion.

Formatting Consistency (FC): Assesses the adherence to syntactic and formatting norms, including punctuation, capitalization, and overall structure. This ensures that the text is readable and conforms to established formatting guidelines.

- **0:** Major formatting issues that significantly impair readability.
- **1-2:** Frequent formatting problems that make the text hard to read or follow.
- **3-4:** Minor inconsistencies that do not substantially hinder readability.
- **5:** Consistent formatting with no issues that impact readability.

Content Relevance (CR): Measures the relevance and informational value of the text, ensuring that it contributes meaningfully to the overall corpus. This criterion evaluates the extent to which irrelevant, extraneous content, or noise is removed.

- **0:** Highly irrelevant, dominated by extraneous or redundant content.
- **1-2:** Some irrelevant material, reducing the value of the text.
- **3-4:** Mostly relevant with occasional minor irrelevant elements.
- **5:** Fully relevant and contributes substantial informational value.

A.2 Annotation Process

The annotation process involves assigning numerical ratings for each of the four criteria, followed by a holistic decision to either *Retain* or *Reject* the sample.

Retain decisions are made for samples that demonstrate sufficient overall quality and educational value, even if minor issues are present. **Reject** decisions are reserved for samples with substantial problems in any criterion or those with minimal educational or informational value.

Disagreements between annotators are resolved by a third expert. If consensus cannot be reached, the sample is excluded from the dataset and replaced with another sample from the same stratification group, ensuring the dataset remains representative of the original distribution.

Annotations are submitted in a structured JSON format. An example is shown below:

```
{
    "SI":        4,
    "CC":        5,
    "FC":        3,
    "CR":        4,
    "Decision": "Retain",
    "Reason":   "Minor format issues
     but overall high educational value
      and relevance."
}
```

These guidelines ensure that the annotation process is rigorous, reproducible, and consistent, allowing for the creation of a benchmark dataset that accurately reflects real-world challenges in pretraining data filtering.

A.3 Data Filtering Track Examples

We annotated a total of 2000 Chinese and 2000 English samples, of which 733/267 were retained/rejected in Chinese and 596/404 in English. Figure 6 presents representative examples illustrating the annotation outcomes.

The first sample (id: `en-filter-186`) is retained due to its clear relevance, high-quality formatting, and educational value despite minor formatting issues. In contrast, the second sample (id: `en-filter-15`) is rejected because it primarily contains raw sports score data with minimal instructional value, limited context, and substantial formatting noise.

B Details in Synthetic Sample Generation

This section provides implementation details for the construction of our synthetic dataset. We describe the prompt templates, corruption functions, and challenge definitions used to generate realistic and diverse samples in both English and Chinese.

B.1 Prompt Templates

To enable fine-grained control over content type and corruption patterns, we adopt a unified prompt template with multiple placeholders. Each placeholder is instantiated using predefined sets drawn from source types, topic hierarchies, and data processing functions. The unified prompt template, shown in Figure 7, guides GPT-4o to generate corrupted but semantically coherent data that reflects the desired complexities.

The placeholder fields, *{source}*, *{sub_source}*, *{topic}*, and *{sub_topic}*, are instantiated according to controlled sampling distributions. As illustrated in Figure 1a and Figure 1b, the inner rings represent high-level categories (data sources and topics), while the outer rings represent their respective subcategories. The selection of data sources follows the proportions shown in Figure 1a(a), whereas sub-sources, topics, and sub-topics are sampled uniformly within each parent category.

In the unified prompt template (Figure 7), all placeholders are rendered in blue to clearly indicate the fields subject to instantiation during generation. These placeholders allow explicit control over the content source and topic scope, enabling the creation of diverse yet structurally aligned data samples.

B.2 Data Processing Functions

Table 4 presents a structured inventory of 30 data processing functions, consolidated from open-source datasets and industrial toolkits. These are grouped by complexity into:

- **Basic:** superficial noise, common tokenization artifacts, spacing.
- **Moderate:** entity distortion, nested formats, mixed script handling.
- **Advanced:** structural reordering, embedded metadata, OCR-like errors.

Each entry includes a function name and a concise description of its specific operations. These entries provide the value sets for the placeholders *{process_func}* and *{func_detail}* used in our prompt templates, ensuring controllable and diverse corruption patterns during synthetic data generation.

```
 {
    "id":   "en-filter-186",
    "text": "The Donaldson Adoption Institute\nSenior Research Fellows\nIssue Areas & Publications\
    nThe Lynn Franklin Fund\nUNIQUE PROJECT AIMS TO PROMOTE POSITIVE IDENTITY FOR TRANSRACIAL
    ADOPTEES\n06/14/2010\tMedia Advisory: For Immediate Release\nNEW YORK, June 14, 2010  The Evan B.
     Donaldson Adoption Institute today announced that it is partnering with two major adoption
    placement and support organizations, Lutheran Social Services of New England and the California-
    based Kinship Center, to develop innovative materials and resources designed to promote positive
     identity formation for transracially adopted children and youth. As a first step in shaping the
     project, the three partners will convene accomplished professionals with experience and
    expertise in this area for a think tank conclave at Tufts University on Wednesday, June 16.\nThe
     project grows out of recommendations included in a groundbreaking study published by the
    Adoption Institute last November entitled Beyond Culture Camp: Promoting Healthy Identity in
    Adoption. The Executive Summary and full study are available, at no charge, on the Institutes
    website at http://adoptioninstitute.org/publications/beyond-culture-camp-promoting-healthy-
    identity-formation-in-adoption.",
    "decision": "Retain"
},
{
    "id":   "en-filter-15",
    "text": "Mount Aloysius vs Penn State Altoona (Apr 1, 2014)\n2014 Penn State Altoona Baseball\
    nMount Aloysius at Penn State Altoona (Game 1)\nApr 1, 2014 at Altoona, Pa. (Stewart Athl. Field
    )\nMount Aloysius 4 (5-10,1-0 AMCC) Penn State Altoona 1 (5-13,0-1 AMCC)\nIan Helsel 2b
    ............. 3 1 2 0 Nate Bennett 2b........... 3 0 0 0\nBryn Brown lf............. 4 0 1 1
    Jordan Swope cf........... 3 0 0 0\nConnor Bowie rf........... 4 1 2 0 Jack Cleary 3b
    ............ 3 0 0 0\nBenjamin Legath dh......... 3 1 2 0 Logan Madill lf........... 3 0 0 0\
    nBrandon Rauhauser pr...... 0 1 0 0 Pat McHenry p............. 2 1 1 0\nCory Dick 3b
    .............. 3 0 1 0 Jared Dailey p........... 0 0 0 0\nJesse Bortner 1b.......... 3 0 2 1
    Jeff Deveney 1b........... 3 0 1 0\nPatrick Gully c........... 4 0 1 0 Nolan Spahr pr
    ........... 0 0 0 0\nDylan Oswalt cf........... 4 0 0 0 John Szmed c.............. 2 0 0 0\
    nBrock Gladfelter ss........ 4 0 0 0 Matt Roland rf............ 2 0 0 0\nDerrick Capiak p
    .......... 0 0 0 0 Joey Myers ph............ 1 0 0 0\nDan Bier ss............... 3 0 0 0\
    nMount Aloysius...... 200 010 1 - 4 11 3\nPenn State Altoona.. 000 010 0 - 1 2 5\nE - Helsel;
    Gladfelter; Capiak; Bennett; Madill; Szmed; Roland; Bier. DP - Altoona\n1. LOB - Mt. Aloysius 11;
     Altoona 5. 2B - Helsel. HBP - Bortner; McHenry; Szmed.",
    "decision": "Reject"
}
```

Figure 6: Examples of annotated data samples. The first sample (ID: en-filter-186) is retained due to its high relevance and clarity, despite minor formatting issues. The second sample (ID: en-filter-15) is rejected due to its low content relevance and poor suitability for model training. These cases exemplify the diversity of data encountered and how annotation criteria are applied in practice.

Table 4: Detailed Informtion of 30 Data Processing Functions by Complexity Level, Function Type, and Specific Operations.

Level	Function	Description and Specific Operations
Basic	Remove Sensitive Info	Eliminate personally identifiable information (e.g., Email, Phone Numbers, Social Security Numbers, Addresses).
	Remove Advertisements	Remove all promotional content from the dataset.
	Remove Harmful Content	Identify and eliminate offensive or harmful content (e.g., Hate Speech, Violent Content, Politically Sensitive Content).
	Remove Headers/Footers	Eliminate extraneous header and footer text commonly found in documents or web pages.
	Standardize Formatting	Normalize text formatting (e.g., Dates, Numbers, Units) for consistency.
	Remove Noise	Remove irrelevant characters, corrupted content, or duplicate text.
	Remove Empty Lines	Eliminate redundant empty lines or excessive breaks in the text.
	Trim Whitespace	Strip leading and trailing whitespace.
	Convert Case	Standardize text to lowercase or uppercase.
	Remove Stopwords	Filter out common words that do not add semantic value.
Moderate	Mask Sensitive Info	Replace sensitive data with placeholders (e.g., Phone Number Placeholders, SSN Placeholders).
	Remove Embedded Links	Eliminate hyperlinks, including anchor text and target URLs.
	Normalize Punctuation	Standardize punctuation based on language-specific conventions.
	DeDuplicate Content	Detect and remove repeated sentences or paragraphs.
	Handle Multilingual Text	Identify and segregate content in mixed-language documents.
	Extract Main Content	Extract core content while discarding non-essential elements.
	Remove Complex Ad Structures	Handle dynamic ad structures, such as nested or script-injected ads.
	Clean OCR Text	Correct recognition errors from Optical Character Recognition (OCR).
	Correct Spelling/Grammar	Fix common spelling and grammatical errors.
	Segment and Label Sections	Use structural markers to divide and label content logically.
Advanced	Handle Contextual Dependencies	Resolve ambiguities requiring deep contextual understanding, such as pronoun references.
	Detect Deeply Embedded Info	Identify sensitive or hidden data embedded within code or HTML.
	Handle Mixed Encoding	Address issues arising from text in multiple character encodings.
	Restore Corrupted Content	Reconstruct damaged or incomplete sections of text.
	Identify and Correct Ambiguity	Resolve unclear references in text.
	Detect Subtle Bias	Neutralize subtle biases (e.g., Gender or Cultural Bias).
	Reformat Tables and Lists	Ensure tables and lists are well-organized and consistent.
	Detect Non-Standard Abbreviations	Expand or clarify uncommon abbreviations.
	Handle Legal Compliance	Ensure processing adheres to privacy laws and copyright regulations.
	Handle Sensitive Cultural Content	Identify and mitigate culturally sensitive material.

B.3 Definitions of Advanced Challenges

To introduce additional difficulty and realism, 50% of samples are assigned to one or both of the following challenge categories:

- **Character Corruption Challenges:** These refer to character-level or formatting-level corruptions that often appear in web-scale corpora but are difficult to detect or normalize. Examples include character substitutions (e.g., "0" vs "O"), irregular spacing, multilingual entanglement (e.g., English-Chinese mixing), malformed lists, or nested quotation structures.

- **Edge Case Challenges:** These reflect edge

Prompt Template for Sample Synthesis

Task: You are required to generate two raw data entries, one in English and one in Chinese, simulating articles from a specified data source and topic. These raw samples should contain natural noise commonly observed in real-world datasets, such as typographical errors, encoding inconsistencies, or layout artifacts. Each type of noise should be recoverable using a corresponding process function, which cleans the data and restores it into a high-quality format suitable for pre-training.

Task Details:

- **Data Source:** The entries should resemble articles originating from {source}, specifically from {sub_source}.

  (e.g., Web pages – online forums; Research papers – scanned PDFs)

- **Topic:** The content of the data entries should reflect the topic of {topic}, and more specifically, focus on {sub_topic}.

  (e.g., Topic: Science & Technology; Sub-topic: Artificial Intelligence in healthcare)

- **Process Function:** The noisy patterns within the data entries can be resolved using {process_func}, which is designed to address {func_detail}.

  (e.g., Function: OCR Error Correction; Detail: Fixes character-level distortions in scanned text)

Ensure the Data Includes:

- **Source- and Topic-Specific Adaptation:** The data should reflect the characteristics of the given source and topic.
- **Content Length:** Both English and Chinese samples must be **at least 500 words long**.
- (optional) **Character Corruption Challenges:** Include complex elements to mimic real-world data challenges:
  - Character Substitutions (e.g., "0" vs "O", "l" vs "I")
  - Spelling Errors, Typos, and Code-Natural Language Mixing
  - Irregular Spacing, Font Distortions, Nested Quotes, and Multi-Level Lists
  - Mixed Elements like page numbers, footnotes, and headers
- (optional) **Edge Case Challenges:** Add scenarios that increase the difficulty for pre-processing:
  - Hidden Sensitive Information (e.g., obfuscated emails, encoded URLs)
  - Encoding Inconsistencies (e.g., UTF-8/GBK mix), hidden text, and watermark content
  - Edge Cases such as extreme text lengths, special characters, or punctuation-only samples

Output Format:
Your output should strictly follow the format below:

```
##EN_RAW##
<English raw data sample>

##ZH_RAW##
<Chinese raw data sample>
```

Final Instructions:

- Only return the **data samples** in the specified format.
- Do **not** include any extra instructions or explanations.
- Ensure that the samples meet the criteria listed above.

Figure 7: Prompt for generating structured data samples with pre-processing functions, focusing on complexities and challenges in the data.

cases or hidden artifacts that require contextual awareness or special handling. Examples include encoded personal information (e.g., Base64 email addresses), mixed encoding formats (e.g., UTF-8 and GBK), extremely short or excessively long sequences, and visually embedded elements like watermarks or hidden text.

The two optional fields in blue in Figure 7 correspond to the inclusion of the above challenge types. We introduce them with controlled probabilities: 20% for *Character Corruption Challenges*, 20% for *Edge Case Challenges*, and 10% for both. This setup introduces additional variability and better reflects the types of difficulties encountered in real-world pretraining corpora.

B.4 Data Cleaning Track Examples

In the data cleaning track of the DATACURBENCH, a total of 6,000 examples were generated, including 3,000 Chinese and 3,000 English samples. These examples are designed to test the model's ability to clean and process noisy text. The figure below presents a representative example from this track, showing both the raw text and the corresponding cleaned text. In this example, the "cleaned_text_human" refers to the human-annotated cleaned text, while the "cleaned_text_reference" represents the cleaned text produced by the GPT-4 model. Additionally, the figure provides meta-information such as the topic, subtopic, source, function, and challenge type, offering a comprehensive view of the cleaning task.

C Prompts for Evaluation

This section outlines the generation and evaluation prompts utilized in the Data Filtering and Data Cleaning tracks, which are central to the curation of high-quality pre-training corpora. These prompts aim to guide both annotators and evaluation models in making systematic judgments on raw and cleaned text samples. The evaluation process is grounded in four primary criteria: sensitive information safety, content relevance, clarity and integrity, and formatting consistency. By employing these structured evaluation criteria, we aim to measure the effectiveness of different models in processing text and identify their strengths and weaknesses, ultimately fostering the creation of robust datasets for further model development.

```
{
    "idx": "en-clean-3753",
    "raw_text": "Chapter 3: Disease Pr3venti0n Techniques  \n\"Dis-ease\" prev3ntion is cr1tical - In
     this l@st book of the series, we focus on vari0us methods to stay healthy. First, let's discuss
     balanced d1ets. A well-planned d1et should include v1tamins, minerals, and pr0teins... Healthy!
     eating habits can greatly reduce the risk of gett1ng sick. **Exercise** is another crucia1
    factor. Regular activ1ty helps maintain weight and enh@c3s immune function. Remember: \"A st1tch
     in time saves nine!\" Pages: 207 \nNotes: [Ensure 50% of diet is fru1ts & vegetab1es]  \
    nFo0tnote: For further information, see \"Chapter 5: Nutrit10n\". \n\nEmail:
    health00xample@encrypted.com  \nImage: [Watermarked with \"Health&Medicine\"]  \nURL: Https://
    prevention-book.eXample.com/ch3 \n\n1. D1sease Causes  \n   - Bacteria  \n   - Vi1use5  \n     *
     Examples include influenza and COVID-19.\n2. Prev3ntion Tips  \n   - Hand washing  \n   -
    Vacc1nations",
    "cleaned_text_human": "Disease prevention is critical. In this section, we focus on methods to
    stay healthy, emphasizing balanced diets that include vitamins, minerals, and proteins. Healthy
    eating habits can reduce sickness risk, while regular exercise maintains weight and enhances
    immune function. Bacteria and viruses cause diseases; prevention includes hand washing and
    vaccinations.",
    "cleaned_text_reference": "Disease prevention is critical. In this section, we focus on methods
    to stay healthy, emphasizing balanced diets that include vitamins, minerals, and proteins.
    Healthy eating habits can reduce sickness risk, while regular exercise maintains weight and
    enhances immune function. Bacteria and viruses cause diseases; prevention includes hand washing
    and vaccinations.",
    "meta": {
          "topic": "Health & Medicine",
          "source": "Books",
          "function": "ExtractMainContent",
          "subtopic": "Disease Prevention",
          "challenge": 2,
          "explanation": "Focus on extracting core content from documents or web pages, discarding
    non-essential elements.",
          "detail_source": "Fiction",
          "detail_function": "Discard Non-Essential Elements"
    }
}
```

Figure 8: Representative example from the data cleaning track, showing raw text and the corresponding cleaned text. The 'cleaned_text_human' represents the human-annotated cleaning result, while 'cleaned_text_reference' refers to the GPT-4 model's annotated cleaning result. Additional meta-information, including topic, subtopic, source, detail source, function, detail function, and challenge type, is also provided.

C.1 Generation Prompt for Data Filtering Track

The Data Filtering Track is designed to assess whether raw text samples should be retained or discarded based on their overall quality. A structured evaluation prompt, as illustrated in Figure 9, guides annotators or evaluation models to make decisions based on four essential criteria: sensitive information safety, content relevance, clarity and integrity, and formatting consistency. These criteria ensure that the resulting pre-training corpus is both high-quality and aligned with ethical standards, minimizing the inclusion of inappropriate or irrelevant content.

C.2 Generation Prompt for Data Cleaning Track

Similar to the Data Filtering Track, the Data Cleaning Track evaluates raw text samples, but its goal is to refine and clean these samples to meet quality standards. The cleaning process follows the same four evaluation criteria, sensitive information safety, content relevance, clarity and integrity, and formatting consistency, ensuring that sensitive content is properly addressed, irrelevant or unnecessary data is removed, and the text maintains its clarity and integrity. The generation prompt, illustrated in Figure 10, instructs the model to perform these tasks with a focus on maintaining consistency and standardization in the cleaned output, making it suitable for further processing in training pipelines.

C.3 Evaluation Prompt for Data Cleaning Track

The Evaluation Prompt for the Data Cleaning Track is used to compare the cleaning effectiveness of different models, specifically the reference model and various evaluation models, shown in Figure 11. The evaluation process involves judges (models) comparing the cleaned outputs generated by the evaluation models against those produced by the reference model. By doing so, the evaluation prompt helps identify the differences in cleaning capabilities between the models. The four key criteria—sensitive information safety, content relevance, clarity and integrity, and formatting consistency—are used to assess the relative quality of the cleaned text. Through these comparisons, we can determine how well the evaluation models perform relative to the reference model and identify areas for improvement in the cleaning process.

Generation Prompt for Data Filtering Track

Task: You are a data filtering expert tasked with evaluating the overall quality and informational value of raw text data intended for LLM pre-training.

Objective: Make a holistic judgment to **retain** data with acceptable overall quality and meaningful content, even if minor issues are present, and to **reject** data with significant flaws or low value.

Evaluation Criteria:

- **Sensitive Information Safety(SI):** This criterion measures the presence of sensitive or harmful content, such as personal information, offensive language, or other material that could pose a privacy risk.

  Reject entries containing substantial personal information, hate speech, or harmful content. Minor residual issues may be tolerated if the content is otherwise valuable.

- **Content Relevance (CR):** Measures the relevance and informational value of the text, ensuring that it contributes meaningfully to the overall corpus. This criterion evaluates the extent to which irrelevant, extraneous content, or noise is removed.

  Retain text with educational or informational value, especially content introducing key concepts, tutorials, or textbook-style explanations. Slight irrelevance or disorganization is acceptable.

- **Content Clarity and Integrity (CC):** Evaluates whether the core meaning and key information of the text are preserved, ensuring that the text is free from significant ambiguity, factual distortion, or loss of meaning.

  Keep entries where the core message is clear and meaningful, even if some ambiguity or omission exists. Discard content that suffers from serious degradation of meaning or logic.

- **Formatting Consistency (FC):** Assesses the adherence to syntactic and formatting norms, including punctuation, capitalization, and overall structure. This ensures that the text is readable and conforms to established formatting guidelines.

  Ensure the text is readable and reasonably structured. Minor formatting problems (e.g., extra whitespace) can be corrected in later processing.

Final Decision Logic:

- **Retain:** If the text has sufficient value for LLM training and only minor imperfections.
- **Reject:** If the text exhibits serious flaws in content, integrity, safety, or relevance.

Required Input/Output Format:

```
Input:
<text>text to be assessed</text>
Output:
<answer>Retain/Reject</answer>
```

Instructions:

- Only return the answer in the required format.
- Do not add any extra commentary or explanations.
- Be strict yet pragmatic in your judgment.

Examples: [Include benchmark examples here for demonstration.]
Actual Input:

```
<text>{raw_text}</text>
```

Expected Output:

```
<answer>
```

Figure 9: Evaluation prompt for the Data Filtering Track, guiding retain/reject decisions based on overall quality and informational value, including safety, clarity, relevance, and formatting.

Generation Prompt for Data Cleaning Track

Task: You are a data cleaning expert responsible for preparing raw text data by removing unwanted or unnecessary elements that may hinder model training. Your goal is to ensure the data is clean, free of irrelevant content, and formatted properly for downstream processes.

Objective: The task is to **remove** any extraneous, irrelevant, or sensitive content from the raw text while **standardizing the format** to make it suitable for large language model (LLM) training. This includes eliminating sensitive personal information, advertisements, irrelevant symbols, and unnecessary noise, ensuring that the text is clear, concise, and well-structured.

Specific Requirements:

- **Sensitive Information Safety (SI):** This criterion measures the presence of sensitive or harmful content, such as personal information, offensive language, or other material that could pose a privacy risk.

  Remove entries containing sensitive data like personal information (e.g., email addresses, phone numbers, physical addresses), offensive language, or harmful content. Ensure no personally identifiable information or privacy risks remain in the cleaned data.

- **Content Relevance (CR):** Measures the relevance and informational value of the text, ensuring that it contributes meaningfully to the overall corpus. This criterion evaluates the extent to which irrelevant, extraneous content, or noise is removed.

  Ensure that all irrelevant or extraneous content, including advertisements, noise, or non-informative sections, is removed. Only the meaningful content should remain.

- **Content Clarity and Integrity (CC):** Evaluates whether the core meaning and key information of the text are preserved, ensuring that the text is free from significant ambiguity, factual distortion, or loss of meaning.

  Remove any extra spaces, improper capitalization, and inconsistent punctuation. Ensure the text is clean, consistent, and properly structured with standardized formatting.

- **Formatting Consistency (FC):** Assesses the adherence to syntactic and formatting norms, including punctuation, capitalization, and overall structure. This ensures that the text is readable and conforms to established formatting guidelines.

  Remove any unnecessary characters, extra spaces, and formatting noise, ensuring the text is clean and easy to process further.

Required Input/Output Format:

```
Input:
<raw_text>text to be processed</raw_text>
Output:
<clean_text>processed clean text</clean_text>
```

Instructions:

- Only return the processed text in the required format.
- Do not include any extra commentary or explanations.
- Strictly adhere to the text cleaning standards.

Examples: [Include benchmark examples here for demonstration.]
Actual Input:

```
<raw_text>{raw_text}</raw_text>
```

Expected Output:

```
<clean_text>
```

Figure 10: Structured prompt used in the Data Cleaning Track Generation task. It guides the transformation of raw noisy text into safe, clean, and well-formatted content for LLM pre-training.

Evaluation Prompt for Data Cleaning Track

Task: You are an expert in text data processing, tasked with evaluating and comparing the quality of processed text generated by two models. Your goal is to assess which model output better aligns with high-quality standards for language model training data.

Specific Requirements:

- **Removal of Sensitive and Harmful Information (SI):** This criterion assesses the extent to which each model output successfully eliminates sensitive content, including personally identifiable information (e.g., names, emails), advertisements, and harmful or offensive language.

  Compare the two outputs and determine which model more thoroughly and reliably removes sensitive or harmful elements. Give preference to the model that leaves no trace of privacy risks or inappropriate content.

- **Content Integrity and Coherence (CC):** This evaluates whether the core meaning, factual information, and logical flow of the original text are preserved during the cleaning process.

  Identify which model better retains the key ideas and expresses them naturally and coherently. Avoid outputs that distort the original meaning or introduce ambiguity.

- **Formatting Consistency (FC):** This measures how well each model enforces consistent formatting, including proper punctuation, spacing, capitalization, and structural clarity.

  Judge which model produces cleaner and more standardized formatting. Favor the output that shows fewer inconsistencies and better aligns with formatting norms.

- **Elimination of Irrelevant Information (CR):** This assesses each model's ability to remove irrelevant or noisy content, such as decorative symbols, unrelated sections, or visual clutter.

  Compare the outputs and select the one that more effectively filters out non-essential content while preserving informative and relevant parts.

Instructions:

- Evaluate each output per the above criteria.
- Select the model that performed best on each criterion.
- Provide a summary of which model generated the highest quality output based on the evaluation criteria.
- **Output Format:** Provide your answer in the following JSON format:

  ```
  {
      "SI": "[model_identifier]",
      "CC": "[model_identifier]",
      "FC": "[model_identifier]",
      "CR": "[model_identifier]",
      "Overall": "[model_identifier]"
  }
  ```

  Only use the identifiers **"A"** or **"B"** in place of [model_identifier], with no additional characters.

Input for Model A and Model B:

```
<raw_text>{raw_text}</raw_text>
```

Model A's Output:

```
<clean_text>{clean_text_a}</clean_text>
```

Model B's Output:

```
<clean_text>{clean_text_b}</clean_text>
```

Evaluation Results:

Figure 11: Prompt for evaluating raw text data for LLM pre-training, based on value, integrity, safety, and formatting.

D Detailed Evaluation Results

D.1 APIs and Models

To ensure reproducibility, Table 5 lists each evaluated proprietary API and open-source model. Proprietary systems are identified by their API version; open-source models are listed with their Hugging Face organization, enabling access through `https://huggingface.co/{org}/{model}`.

Table 5: API Versions for Proprietary Systems and Hugging Face Organizations for Open-Source Models

Model Name	Version / HF Org.
GPT-4o	GPT-4o-2024-08-06
GPT-4	GPT-4-1106-preview
Gemini	Gemini-1.5-Pro
Claude-3.5	Claude-3.5-Sonnet
gemma-2-2b	google
gemma-2-2b-it	
gemma-2-9b	
gemma-2-9b-it	
Llama-3.2-1B	meta-llama
Llama-3.2-1B-Instruct	
Llama-3.2-3B	
Llama-3.2-3B-Instruct	
Meta-Llama-3-8B	
Meta-Llama-3-8B-Instruct	
MiniCPM-1B-sft-bf16	openbmb
MiniCPM-2B-sft-bf16	
MiniCPM3-4B	
Mistral-7B-v0.3	mistralai
Mistral-7B-Instruct-v0.3	
Phi-3-mini-128k-instruct	microsoft
Phi-3-mini-4k-instruct	
Phi-3-small-128k-instruct	
Phi-3-small-8k-instruct	
phi-1_5	
phi-2	
Qwen2.5-0.5B	Qwen
Qwen2.5-0.5B-Instruct	
Qwen2.5-1.5B	
Qwen2.5-1.5B-Instruct	
Qwen2.5-3B	
Qwen2.5-3B-Instruct	
Qwen2.5-7B	
Qwen2.5-7B-Instruct	
Yi-1.5-9B	01-ai

D.2 Cleaning Performance Across Different Dimensions

Table 6 reports the full win rate results for each evaluated model across the four dimensions of the cleaning task, Sensitive Information Safety(SI), Content Clarity and Integrity (CC), Formatting Consistency (FC), and Content Relevance (CR), as well as the overall win rate.

Table 6: Model performance across different dimension of evaluation criteria.

Model	Overall	SI	CC	FC	CR
gemma-2-2b	26.57	24.12	37.92	34.73	21.43
gemma-2-2b-it	31.77	29.68	41.52	38.55	27.92
gemma-2-9b	32.37	35.10	39.28	34.58	34.35
gemma-2-9b-it	44.53	46.55	48.60	42.17	45.50
Llama-3.2-1B	18.35	19.93	26.38	24.48	16.90
Llama-3.2-1B-Instruct	19.90	24.08	25.82	26.92	23.02
Llama-3.2-3B	23.67	26.83	30.23	27.72	26.53
Llama-3.2-3B-Instruct	30.40	33.53	36.07	33.00	30.00
Meta-Llama-3-8B	31.45	28.63	42.48	37.93	26.17
Meta-Llama-3-8B-Instruct	34.90	49.90	31.22	34.00	54.48
MiniCPM-1B-sft-bf16	19.40	25.12	25.87	25.25	23.58
MiniCPM-2B-sft-bf16	7.23	11.43	9.33	9.13	9.68
MiniCPM3-4B	29.40	35.22	35.17	32.10	35.05
Mistral-7B-Instruct-v0.3	38.78	47.62	36.82	37.22	51.95
Mistral-7B-v0.3	32.68	30.70	41.70	37.82	29.07
phi-1_5	13.98	17.50	21.15	20.98	14.27
Phi-3-mini-128k-instruct	31.40	46.88	26.62	31.62	51.33
Phi-3-mini-4k-instruct	30.73	48.00	24.97	30.60	53.75
Phi-3-small-128k-instruct	36.25	49.70	32.93	33.37	51.37
Phi-3-small-8k-instruct	31.28	50.08	27.02	29.05	52.88
Phi_2	23.62	22.67	34.65	32.17	19.45
Qwen2.5-0.5B	16.55	17.60	24.73	23.78	14.80
Qwen2.5-0.5B-Instruct	18.78	24.05	22.65	26.73	22.43
Qwen2.5-1.5B	25.12	23.00	36.18	34.05	20.50
Qwen2.5-1.5B-Instruct	29.55	29.32	33.48	36.17	31.38
Qwen2.5-3B	28.43	29.70	36.67	34.37	28.38
Qwen2.5-3B-Instruct	34.57	33.50	40.72	38.47	33.70
Yi-1.5-9B	19.22	40.22	15.83	19.75	49.22
MiniCPM-1B-sft-bf16	21.27	28.48	27.45	28.08	35.95
MiniCPM-2B-sft-bf16	23.47	25.93	37.02	31.17	25.65
MiniCPM3-4B	31.04	37.67	36.89	33.39	41.80
Qwen2.5-0.5B	18.32	18.32	20.75	33.43	26.17
Qwen2.5-0.5B-Instruct	18.38	22.45	22.45	24.18	26.82
Qwen2.5-1.5B	28.45	28.95	28.95	40.68	35.35
Qwen2.5-1.5B-Instruct	36.83	36.83	35.87	46.62	43.78
Qwen2.5-3B	23.03	21.75	40.52	31.77	19.48
Qwen2.5-3B-Instruct	26.67	28.37	34.82	34.70	32.58
Qwen2.5-7B	33.52	38.20	39.47	36.80	40.25
Qwen2.5-7B-Instruct	42.40	46.50	44.17	41.93	50.53
Yi-1.5-9B	23.13	36.23	20.57	25.20	56.68

D.3 Additional Results on In-Context Example Effectiveness

To complement the main results presented in Table 1, we report additional experimental findings on the impact of in-context examples across different model families and configurations in Table 7. These include performance metrics on both the cleaning and filtering tracks for various base and instruction-tuned models, covering English and Chinese subsets. The extended results further support the observed trends regarding model size, instruction tuning, and the efficacy of in-context prompting, particularly for smaller or less instruction-aligned models.

Table 7: Additional 6-shot vs. 0-shot comparison; bold indicates rare 0-shot wins

Model	Cleaning		Filtering	
	6-shot	0-shot	6-shot	0-shot
EN Subset				
Phi-3-small-128k-Instruct	36.65	34.38	22.03	9.84
Phi-3-small-8k-Instruct	31.28	**33.85**	38.42	13.76
Phi-3-mini-128k-Instruct	31.40	**37.07**	34.52	**38.98**
Phi-3-mini-4k-Instruct	30.73	**35.23**	20.56	16.18
phi_2	23.62	**24.65**	34.51	34.13
phi-1_5	13.98	7.65	6.12	0.00
Mistral-7B-Instruct-v0.3	38.78	37.23	22.32	11.57
Mistral-7B-v0.3	32.68	31.45	6.59	0.50
MiniCPM3-4B	29.40	29.38	59.85	23.50
MiniCPM-2B-sft-bf16	7.23	**20.62**	36.30	20.35
MiniCPM-1B-sft-bf16	19.40	**20.25**	49.87	7.87
Yi-1.5-9B	19.22	**27.25**	40.90	12.19
ZH Subset				
MiniCPM3-4B	31.04	29.62	37.39	1.49
MiniCPM-2B-sft-bf16	23.47	19.63	48.14	17.53
MiniCPM-1B-sft-bf16	21.27	14.48	44.59	13.38
Yi-1.5-9B	23.13	**27.35**	47.78	3.61