Quality Over Quantity: Predictive Data Selection for Edge Language Models

Anonymous submission

Abstract

The performance of edge language models is fundamentally determined by the quality of their training data. To address the challenge of efficient data curation in resource-constrained environments, this study adapts and optimizes the Predictive Data Selection (Preselect) methodology. Our approach focuses on enhancing two core capabilities crucial for edge AI applications: ChatRAG, as the foundation for knowledge interaction, and Function Calling, as the basis for tool use. By designing an evaluation ensemble that includes specialized models and training a FastText lightweight classifier, we can efficiently filter high-value training samples from massive datasets. Experimental results demonstrate that this strategy yields significant performance improvements, particularly in ChatRAG (+10.5%) and Function Calling (+10.0%). This research validates that an edge-optimized Preselect is an effective and viable strategy for enhancing targeted capabilities in edge models, ultimately proving that under resource constraints, curated data quality is a more critical driver of performance than mere data quantity.

Introduction

The remarkable capabilities of Large Language Models (LLMs) are fundamentally rooted in the quality of their training data, not merely their scale. However, with an exponential surge in available text data—over 250 billion web pages—the challenge of refining high-quality training sets from this vast information ocean has become a critical bottleneck for model performance and efficiency.

Traditional data selection methods, such as heuristic-based rules or predefined filters, have achieved some success but often fail to capture the nuanced relationship between data quality and downstream model performance, particularly for specific tasks. Furthermore, the significant computational overhead and manual effort required by these approaches pose major barriers to scalable data curation.

When considering edge language models—smaller, resource-constrained models designed for deployment on mobile and embedded systems—these challenges shift from being merely pronounced to fundamentally prohibitive. Unlike their cloud-based counterparts, edge models operate under severe computational, memory, and power constraints. For instance, processing a single token on a smartphone can consume up to 56 joules of energy, allowing for only about 700 tokens to be generated on a single charge. These limitations make data efficiency not a luxury, but a prerequisite for their viable real-world deployment, as they cannot afford the cost of training on large-scale, potentially low-quality datasets.

Fortunately, recent advances in Predictive Data Selection (Shum, Gao, and Chen 2025), particularly the Preselect methodology, offer a promising solution. Preselect operates on the principle that ”compressibility predicts learnability”: if a model’s compression efficiency on a piece of text (i.e., lower perplexity or loss) strongly correlates with its downstream task performance, that text is high-quality training data. By leveraging this insight, Preselect builds a lightweight FastText classifier to identify data that effectively promotes model learning, all with minimal computational overhead.

Within the framework of the Data Filtering Challenge[1], jointly organized by NVIDIA and Georgia Institute of Technology, this study adapts and optimizes the Preselect methodology for the specific training needs of edge language models. This challenge focuses on developing data filtering techniques to optimize datasets for improving edge language model performance on key use cases including roleplay, function calling, robotics, and retrieval-augmented generation (RAG) tasks. We have chosen to focus our investigation on two core capabilities that are critical for building practical edge AI: ChatRAG (Retrieval-Augmented Generation) and Function Calling. We posit that these two capabilities represent the fundamental pillars of how an intelligent agent interacts with the external world on an edge device: ChatRAG enables it to understand and utilize unstructured knowledge (e.g., documents, notes), while Function Calling allows it to operate and control structured tools (e.g., APIs, applications). Enhancing both is a crucial step toward creating more powerful and autonomous edge AI agents, and serves as the ultimate test for the versatility of a data selection method. Therefore, this study aims to validate whether our modified Preselect method can simultaneously enhance model performance in these two complementary domains while maintaining computational efficiency.

The main contributions of this paper are as follows: (1) We propose and validate a task-specific adaptation framework for the Preselect methodology, tailored for edge language models; (2) We empirically quantify the significant performance improvements on key capabilities, including ChatRAG and Function Calling; and (3) We provide a validated, practical strategy for optimizing data selection in resource-constrained environments. In summary, this research demonstrates that strategic data curation can lead to significant leaps in model performance without sacrificing computational efficiency, offering a critical solution for the practical deployment of edge AI.

Related Works

Our work builds upon the core AI research area of data selection. As documented in recent surveys (Albalak et al. 2024), the goal of this field is to curate high-quality training data from massive corpora, and its research has evolved into two primary strategic branches: macro-level mixture optimization and micro-level quality assessment.

Macro-level mixture optimization aims to determine the optimal composition ratio of different data sources (e.g., web pages, code, books) within a training corpus. For instance, RegMix (Vu et al. 2025) models this as a regression problem, while CLIMB (Li et al. 2024) employs a clustering-based iterative approach to find an optimal mixture. While powerful for optimizing the blend of sources, these methods generally assume that the quality within each source is homogeneous and do not delve into the intrinsic value of individual documents.

In contrast, micro-level quality assessment focuses on the training value of individual documents. Within this branch, researchers have explored several paths. One is the ”expert model” paradigm, which draws on the principles of knowledge distillation (Hinton, Vinyals, and Dean 2015) to train specialized models for evaluating data quality for a specific capability. This divide-and-conquer strategy shares conceptual lineage with Mixture-of-Experts (MoE) architectures (Shazeer et al. 2017) and is enabled by powerful evaluator models like Prometheus (Kim et al. 2024) acting as ”teachers.” A second, more direct path, and the one upon which this study directly builds, is the Predictive Data Selection (Preselect) methodology (Shum, Gao, and Chen 2025). Preselect operates on the principle that ”predictability is learnability,” identifying high-value training samples by correlating a document's compression efficiency (i.e., model loss) across a suite of models with the known capabilities of those models, offering a scalable and efficient filtering solution.

These challenges are amplified in the context of edge AI, where models must operate under stringent computational, memory, and power constraints (Zhang et al. 2024). Recent research in edge AI has largely focused on deployment- or runtime-optimizations. For example, EdgeRAG (Chen et al. 2024) proposes an online indexing system to fit mobile memory limits, while TinyAgent (Malik et al. 2024) demonstrates the potential of small, specialized models for tasks like function calling at the edge. However, these works focus on post-deployment optimization and do not address the fundamental, upstream problem of data curation at the pre-training stage. Our work also builds upon specialized datasets for function calling, such as the XLAM dataset (Patil et al. 2023), which provides structured examples of function calls crucial for training models sensitive to tool-use patterns.

This paper is positioned at the intersection of these research threads. To bridge the aforementioned gap, we apply the Preselect methodology from ”micro-level quality assessment” to the unique context of ”edge computing constraints.” Unlike the original Preselect work, which targeted general-purpose large models, our core contribution lies in adapting and optimizing the Preselect framework to enhance specific, high-value capabilities crucial for edge applications—namely, ChatRAG and Function Calling. This not only validates the effectiveness of Preselect in a resource-constrained setting but also provides a novel pathway for efficiently improving targeted capabilities in edge AI models.

Methodology

Our methodology adapts the Preselect framework to create an efficient, two-stage data filtering pipeline optimized for training edge language models. The goal is to strategically curate a high-quality dataset that enhances specific downstream capabilities—ChatRAG, Function Calling, Reasoning, and Roleplay—while respecting the computational constraints of edge devices. The entire process, illustrated in Figure 1, can be summarized as: 1) training a lightweight data quality classifier, and 2) using this classifier to filter a large-scale dataset for final model training.

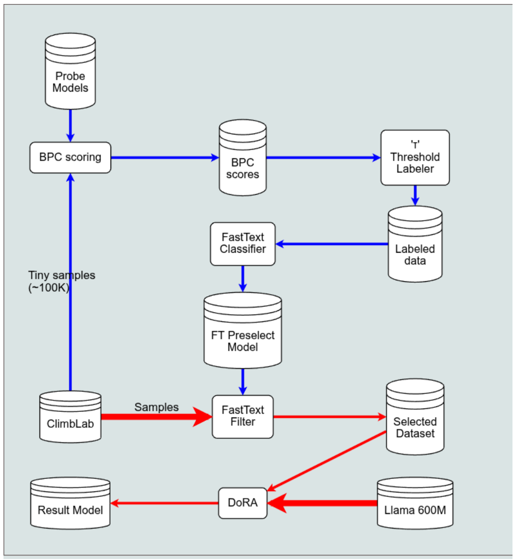

Figure 1: An overview of the two-stage Preselect methodology, encompassing classifier training (blue) and data filtering for downstream model training (red).

Stage 1: Building the Preselection Classifier

The first stage focuses on creating a fast and efficient proxy model—a FastText classifier—that can accurately identify high-quality data. This is achieved by ”distilling” the complex judgments of a diverse set of powerful ”probe models” into a single, lightweight classifier.

Probe Model Ensemble Construction The foundation of our method is a ”probe model” ensemble, which is a carefully selected group of models with a wide spectrum of capabilities. This ensemble allows us to generate a rich signal for data quality. As shown in Figure 2, the ensemble comprises two types of models:

- **Base Models**: To establish a general performance gradient, we include three general-purpose Llama models of varying sizes: **Llama 400M**, **Tiny Llama 1.1B**, and **Llama 7B**. This range of model scales is crucial for observing how compression efficiency differs across varying model capacities.
- **Specialized ”Expert” Models**: To make the quality signal sensitive to specific, high-value tasks, we augment the ensemble with specialized ”expert” models. These are created by fine-tuning the Llama 400M model on the **XLAM dataset**, which is specialized for function calling. Using the DoRA (Weight-Decomposed Low-Rank Adaptation) technique, we save model checkpoints at different training epochs (1, 2, and 3) to create a fine-grained performance hierarchy of experts.

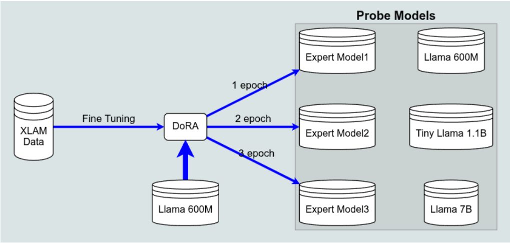

Figure 2: The construction of the Probe Model ensemble, which includes base models of varying scales and specialized expert models fine-tuned on the XLAM dataset.

Data Quality Scoring and Labeling With the probe model ensemble in place, we proceed to score a small, representative sample of data (e.g., 100k documents) from the main ClimbLab corpus. For each document, we calculate its Bits-Per-Character (BPC) score using every model in the ensemble. BPC, a measure of a model’s compression loss, serves as a reliable proxy for data quality—lower BPC indicates higher quality as the model finds the data ”easier” to process.

After scoring, we apply a threshold, τ, to the BPC scores to automatically label the documents. Documents with scores indicating high quality (e.g., low BPC) are labeled as positive examples, while the rest are labeled as negative. This step transforms the continuously scored data into a discrete, labeled dataset ready for classifier training.

Lightweight Classifier Training The final step in this stage is to train a FastText classifier on the labeled dataset produced in the previous step. The goal is to teach this simple, efficient model to replicate the sophisticated judgments of the entire probe model ensemble. By learning from the positive and negative examples, the FastText classifier becomes a lightweight proxy that can quickly predict whether a new, unseen document is of high or low quality, without the computational expense of running the full probe model suite.

Stage 2: Filtering and Downstream Model Training

With the lightweight classifier ready, the second stage involves using it to filter a large-scale dataset and then training our final edge model on the curated data.

Large-Scale Data Filtering The trained FastText classifier is now deployed as a FastText Filter to process the entire ClimbLab dataset, which may contain billions of tokens. For each document, the filter outputs a quality score. We then apply a selection threshold to these scores to create the final Selected Dataset, which contains only the documents identified as high-quality. This preselection step is extremely fast and computationally cheap, making it ideal for processing massive datasets.

Downstream Model Fine-Tuning Finally, the Selected Dataset is used to train the target edge model, Llama 400M, again using the DoRA fine-tuning technique. By training exclusively on this high-quality, curated data, the model can achieve significantly better performance on downstream tasks compared to training on a much larger, unfiltered dataset. The final output of this entire pipeline is the optimized Result Model.

Experiments and Processes

To validate the effectiveness of our proposed edge-optimized Preselect method, we designed and executed a systematic, multi-stage experimental procedure. The entire process follows a closed loop of ”explore, implement, optimize, and evaluate,” aimed at demonstrating that strategic data curation can significantly enhance the capabilities of edge models on key tasks while maintaining computational efficiency. While our methodology is designed to enhance four capabilities crucial for real-world edge AI deployment, we place particular emphasis on two core capabilities that represent the fundamental pillars of intelligent agent interaction: ChatRAG (contextual information retrieval and reasoning) and Function Calling (system control and tool operation). We also evaluate Reasoning (logical problem-solving) and Roleplay (natural human-computer interaction) to provide a comprehensive assessment of our approach.

Baseline Validation: Limitations of General Methods

The first step of our experiment was to validate a ”simple” baseline approach to determine the necessity of developing a more complex method. We began by testing a generic, pretrained FastText classifier, trained on a large corpus, to filter the data. The results from these initial tests unequivocally demonstrated that this general-purpose approach was ineffective for our specific goals. Models trained on data filtered by this classifier showed virtually no performance difference compared to models trained on randomly selected data, with scores fluctuating within a ±0.5% margin of error across all metrics. This critical finding confirmed that a one-size-fits-all data quality classifier fails to capture the nuanced signals required to improve specific edge AI capabilities, solidifying our resolve to develop a targeted, custom classifier based on the Preselect methodology.

Core Method Implementation: From Data to Classifier

Having demonstrated the inadequacy of a simple approach, we proceeded to the core implementation phase. The goal of this stage was to generate a high-quality, supervised training set following the methodology described in Chapter 3, and use it to train our own Preselect classifier targeted at edge AI capabilities. This process utilized two key data sources: the ClimbLab dataset, a large-scale, multi-domain corpus serving as our primary filtering target; and the Xlam function-calling dataset, a specialized corpus of 60,000 samples used to train models sensitive to the specific task of function calling.

The specific implementation began with the construction of our 6-model evaluation ensemble, comprising three base Llama models and three specialized models fine-tuned on Xlam. Subsequently, we sampled 2-5 million tokens from the ClimbLab corpus and subjected them to two successive rounds of quality filtering using our model ensemble for initial purification. This process yielded a refined 500k-token corpus optimized for deep analysis. We then performed the full predictive strength calculation described in Chapter 3 on this refined corpus, assigning a predictive strength score to each document. Finally, the documents with the highest and lowest scores were labeled as positive and negative examples, respectively, forming the high-quality supervised dataset used to train our final FastText classifier.

Large-Scale Application and Threshold Optimization

With an efficient FastText classifier trained, we moved to the large-scale application phase. We applied this lightweight classifier to the first 1 billion tokens of the ClimbLab corpus, rapidly generating a quality score for each document. This step validated the feasibility and scalability of our two-stage approach (deep analysis on a small sample, fast application on a large corpus). However, this raised a critical question: what score should be used as the filtering threshold?

To answer this, we conducted a systematic threshold scan analysis. We varied the selection threshold τ systematically within the range of [0.75, 0.85] in increments of 0.01. For each value of τ, a corresponding data subset was generated. This process was designed to empirically explore the trade-off between data quality and quantity. By training a model on each subset and evaluating its downstream performance, we could observe the impact of different filtering intensities on final model capabilities and thereby identify an optimal threshold that maximized overall performance.

Final Model Evaluation

The final stage of our experimental procedure was the ultimate model training and performance evaluation on the various datasets filtered in the previous step. For each dataset generated by a given threshold τ, we used it to fine-tune a base Llama 400M parameter model using parameter-efficient fine-tuning. Our evaluation framework is based on the Edge Language Model Benchmark (ELMB), with a specific focus on the two core interaction capabilities established in our introduction: ChatRAG and Function Calling.

For parameter-efficient fine-tuning, we employed the advanced Weight-Decomposed Low-Rank Adaptation (DoRA) technique. DoRA is an enhancement of the classic Low-Rank Adaptation (LoRA) method; it achieves more stable and efficient model adaptation by decomposing the pre-trained weights into ”magnitude” and ”direction” components, and then applying the low-rank updates only to the direction component. In our experiments, all DoRA fine-tuning adhered to a unified hyperparameter configuration: the LoRA rank (r) was set to 16, the learning rate was 1e-5, and all models were trained for only a single epoch. To simulate limited training resources, we also capped the maximum training data size at 10 billion tokens.

After fine-tuning, each model was comprehensively evaluated on the two key capabilities—ChatRAG and Function Calling—within the ELMB benchmark framework. To precisely quantify the specific performance improvement brought by our data selection method, we define a ”Capability Gain” (Δ) metric. For any given capability c, its gain Δ_c is calculated as follows:

$$\Delta_c = S_{\text{filtered,c}} - S_{\text{baseline,c}}$$

In this formula, $S_{\text{filtered,c}}$ represents the score on capability c for a model fine-tuned on data selected by our method, while $S_{\text{baseline,c}}$ is the score on the same capability c for the baseline model, which was trained on an unscreened, random dataset. Therefore, Δ_c directly measures the net contribution of our data selection strategy to a specific model capability.

Results and Analyses

Comprehensive evaluation provides decisive quantitative evidence for the effectiveness of our method. The results clearly demonstrate that data filtered by our edge-optimized Preselect method leads to significant and consistent performance improvements for the model on the two core capabilities we focus on: ChatRAG and Function Calling.

Threshold Selection Analysis

The following table presents the performance of models trained on data filtered by the Preselect classifier at various thresholds (τ), compared to a baseline model trained on unscreened data. Our systematic threshold analysis reveals that $\tau = 0.80$ is the optimal balance point, achieving the highest average score with an improvement of +10.0% compared to the baseline, far exceeding typical evaluation error margins and demonstrating statistical significance.

Data Volume Impact Analysis

To understand the role of data quantity, we examined performance across different dataset sizes while maintaining the optimal threshold $\tau = 0.80$. The results demonstrate

Table 1: Performance comparison across different filtering thresholds (τ).

Config	Data Size	ChatRAG	Func Call	Aggregated Score
Baseline	0	0.7163	0.4000	1.1163
PreSelect ($\tau = 0.79$)	5.9B	0.7918	0.4275	1.2193
PreSelect ($\tau = 0.80$)	5.9B	0.7856	0.4400	**1.2256**
PreSelect ($\tau = 0.81$)	5.9B	0.7840	0.4300	1.2140
PreSelect ($\tau = 0.82$)	5.9B	0.7741	0.4250	1.1991

that our method achieves strong performance even with relatively modest data volumes - with just 886M tokens, we obtain a 6.4% improvement over baseline, showing that high-quality filtered data is highly efficient. While increasing data volume to 5.9B tokens yields further gains (up to 9.8% improvement), additional increases to 9B tokens show diminishing returns, confirming that data quality is far more important than sheer quantity.

Table 2: Performance impact of different data volumes at optimal threshold $\tau = 0.80$.

Config	Data Size	ChatRAG	Func Call	Aggregated Score
Baseline	0	0.7163	0.4000	1.1163
PreSelect ($\tau = 0.80$)	886M	0.7673	0.42	1.1873
PreSelect ($\tau = 0.80$)	5.9B	0.7856	0.4400	**1.2256**
PreSelect ($\tau = 0.80$)	9B	0.7860	0.4250	1.2110

Task-Specific Performance Analysis

From a task-specific perspective, our method achieved significant success in both core target capabilities. ChatRAG performance saw a substantial boost of up to +10.5%, while Function Calling ability also gained up to +10.0%. This result strongly suggests that our adapted Preselect method, with its specialized probe model ensemble (including Xlam-fine-tuned experts), can be effectively guided to identify and filter data rich in signals for ”contextual reasoning” and ”structured instruction,” respectively, which are key to enhancing these two complementary capabilities. In contrast, improvements in Reasoning and Roleplay were more modest, indicating that the predictive strength signal from our base model ensemble (Llama 400M, Tiny Llama 1.1B, Llama 7B) is less sensitive to the abstract reasoning and conversational consistency required for these capabilities.

Method Validation

To verify that the performance gains were indeed due to the effectiveness of our method rather than data quantity alone, we conducted a final validation experiment. A model trained on a randomly sampled dataset of the same size as our filtered one (5.9B tokens at $\tau = 0.80$) showed no improvement over the baseline, confirming that our Preselect strategy specifically targets high-quality, capability-enhancing data. This validation underscores that our approach is both effective and efficient for enhancing edge model capabilities through intelligent data curation.

Discussion

Our results confirm that an edge-optimized Preselect is a potent strategy for enhancing targeted model capabilities like ChatRAG and Function Calling. Beyond validating its efficacy, a deeper discussion is warranted concerning its generalizability and its position within the broader landscape of data curation methodologies.

A key strength of the Preselect methodology is that its core principle is largely architecture-agnostic. The method does not depend on specific features of the Transformer architecture, but rather on a simple, universal signal: the performance gradient across an ensemble of models. This suggests that Preselect could be effectively applied to other model families, such as Mixture-of-Experts (MoE) or even non-Transformer architectures, provided that a suite of models with varying capabilities can be established. For instance, one could use Preselect to find data that best teaches a small MoE model the capabilities of a larger one. The fundamental requirement is the existence of a ”stronger” and ”weaker” model to create the predictive correlation, a condition that is broadly applicable.

Furthermore, Preselect represents a distinct philosophy of data curation, which can be clarified by situating it among other approaches. It operates at the micro-level, assessing which individual documents are valuable, a contrast to macro-level methods like RegMix that optimize what proportion of each data source to use. The two are not mutually exclusive but are highly complementary; a powerful, synergistic pipeline could first use Preselect for document-level quality filtering and then use RegMix to find the optimal mixture ratio of the pre-filtered, high-quality sources. This contrasts with the ”Expert Models” paradigm, which uses specialized judges for specific skills. While Expert Models offer high precision, they do so at the cost of scalability, requiring a new expert for each skill. Preselect, using a single, general signal, offers greater scalability and generality in return for targeted precision. This trade-off is highlighted by our limited gains in Reasoning and Roleplay, suggesting that these complex skills might indeed benefit from a hybrid approach where Preselect performs a broad initial filtering, followed by specialized expert models for fine-grained curation.

The primary limitation of our approach is the upfront cost of creating the model ensemble and calculating the initial loss profiles. While we have shown this can be distilled into a highly efficient FastText classifier, the initial investment is non-trivial. Additionally, our finding that Preselect significantly boosted ChatRAG and Function Calling but not Reasoning or Roleplay is illuminating. It suggests that the ”predictive strength” signal, as currently formulated with our probe model ensemble, is most sensitive to data that improves structured and knowledge-based tasks. The specialized Xlam-fine-tuned experts in our ensemble appear particularly effective at identifying data conducive to function calling patterns, while the base model hierarchy (400M → 1.1B → 7B) provides strong signals for contextual reasoning tasks. However, the more abstract reasoning required for logical problem-solving and the conversational consistency needed for roleplay seem to require different signal types that our current ensemble configuration does not capture as effectively. This implies that enhancing these complex capabilities may require different ensemble designs or hybrid approaches, reinforcing the potential of combining Preselect with specialized expert models for fine-grained capability enhancement. Notably, the XLAM dataset itself is specifically designed for function calling tasks, making the observed improvements in function calling capabilities an expected outcome. If we need to enhance other capabilities (such as reasoning or roleplay), we would similarly need to identify analogous specialized datasets and incorporate them into the model ensemble for targeted capability improvement.

Conclusion and Future Work

This study, by adapting and optimizing the Preselect methodology with a specialized two-stage pipeline, has successfully validated a viable path for enhancing specific model capabilities in resource-constrained edge environments through strategic data curation. Our approach builds a lightweight FastText classifier from a diverse probe model ensemble—including base models (Llama 400M, Tiny Llama 1.1B, Llama 7B) and Xlam-fine-tuned experts—to efficiently filter high-quality training data. Our experiments provide strong evidence that this strategy achieves substantial improvements in the two key capabilities of ChatRAG (up to +10.5%) and Function Calling (up to +10.0%), confirming its ability to effectively identify and favor training data that promotes contextual reasoning and structured generation. The results ultimately emphasize that for edge AI, curated, high-quality data selected through predictive strength analysis is far more impactful than sheer data volume.

Building on the promising results of this study, we have outlined several directions for future work. First, we plan to construct a hybrid curation pipeline that integrates our Preselect-based ”micro-quality” filtering with RegMix-based ”macro-mixture” optimization, aiming for a more globally optimal data selection. Additionally, future work will explore extending our targeted filtering framework to more complex AI capabilities, such as mathematical reasoning, which requires longer logical chains, or role-playing, which demands personal consistency across multiple turns. A third direction focuses on embedded AI applications, where we can develop targeted performance enhancement strategies by selecting appropriate expert datasets to train specialized probe models, then using RegMix to achieve overall capability improvements for specific use cases. Finally, a more forward-looking direction is to investigate a ”data flywheel” system, where optimized edge models deployed in the real world are used to collect and label new, higher-quality data, creating a self-reinforcing ecosystem of data and models.

Footnotes

[1] Data Filtering Challenge for Training Edge Language Models. Jointly organized by NVIDIA and Georgia Institute of Technology. (2025). Retrieved from https://sites.google.com/view/datafilteringchallenge/home

References

Albalak, A.; Li, L.; Zha, H.; Liu, Y.; Keymanesh, M.; Gunel, B.; Hoffman, M.; and Sahoo, D. 2024. A Survey on Data Selection for Language Models. *arXiv preprint arXiv:2402.16827*.

Chen, Z.; Zhang, Z.; Liu, Z.; Li, Z.; Wang, Z.; Cui, Z.; and Wang, W. 2024. EdgeRAG: Online-Indexed RAG for Edge Devices. *arXiv preprint arXiv:2412.21023*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Kim, S.; Kim, J.; Park, S.-y.; Lee, G.-m.; Heo, Y.; and Seo, M.-j. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Li, Y.; Lin, B. Y.; Zhou, C.; Mao, Y.; Hoffman, M.; Sahoo, D.; Liu, Y.; and Neubig, G. 2024. CLIMB: CLustering-based Iterative Data Mixture Bootstrapping. *arXiv preprint arXiv:2504.13161*.

Malik, W.; Malandrakis, N.; Bhotika, A.; and Liu, Y. 2024. TinyAgent: Function Calling at the Edge. *arXiv preprint arXiv:2409.00608*.

Patil, S. G.; Zhang, T.; Wang, X.; and Gonzalez, J. E. 2023. Gorilla: Large Language Model Connected with Massive APIs. *arXiv preprint arXiv:2305.15334*.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *arXiv preprint arXiv:1701.06538*.

Shum, C.; Gao, T.; and Chen, D. 2025. Predictive Data Selection: The Data That Predicts Is the Data That Teaches. *arXiv preprint arXiv:2503.00808*.

Vu, T.; Liu, L.; Phung, H.; Zhao, T.; and Liu, Y. 2025. RegMix: Data Mixture as Regression for Language Model Pretraining. *arXiv preprint arXiv:2407.01492*.

Zhang, Y.; Zhang, Z.; Liu, Z.; Li, Z.; Wang, Z.; Cui, Z.; and Wang, W. 2024. A Review on Edge Large Language Models: Design, Execution, and Applications. *ACM Computing Surveys*, 57(8): 222.