Task-Aware Reduction for Scalable LLM–Database Systems

1st Marcus Emmanuel Barnes
Faculty of Information
University of Toronto
Toronto, Canada
marcus.barnes@utoronto

2nd Taher A. Ghaleb
Department of Computer Science
Trent University
Peterborough, Canada
taherghaleb@trentu.ca

3rd Safwat Hassan
Faculty of Information
University of Toronto
Toronto, Canada
safwat.hassan@utoronto.ca

Abstract—**Large Language Models (LLMs) are increasingly applied to data-intensive workflows, from database querying to developer observability. Yet the effectiveness of these systems is constrained by the volume, verbosity, and noise of real-world text-rich data such as logs, telemetry, and monitoring streams. Feeding such data directly into LLMs is costly, environmentally unsustainable, and often misaligned with task objectives. Parallel efforts in LLM efficiency have focused on model- or architecture-level optimizations, but the challenge of reducing upstream input verbosity remains underexplored.**

In this paper, we argue for treating the token budget of an LLM as an *attention budget* and elevating task-aware text reduction as a first-class design principle for language–data systems. We position input-side reduction not as compression, but as attention allocation: prioritizing information most relevant to downstream tasks.

We outline open research challenges for building benchmarks, designing adaptive reduction pipelines, and integrating token-budget–aware preprocessing into database and retrieval systems. Our vision is to channel scarce attention resources toward meaningful signals in noisy, data-intensive workflows, enabling scalable, accurate, and sustainable LLM–data integration.

Index Terms—**Large language models (LLMs), task-aware text reduction, log analysis, sustainable AI, Continuous Integration (CI), log reduction, information retrieval and database integration**

I. INTRODUCTION

Large Language Models (LLMs) are increasingly embedded into data-intensive workflows, powering natural language interfaces for databases, observability platforms, and developer tools [1, 2]. They enable capabilities such as conversational querying, automated debugging, and intelligent system triage, transforming how users interact with large-scale, heterogeneous data [3, 4]. As organizations adopt LLMs at the core of data management and software engineering processes, the integration of language and data has become both a research challenge and a practical necessity.

However, the effectiveness of LLMs in these settings is constrained by real-world data. Logs, telemetry streams, and execution traces are verbose, noisy, and dominated by task-irrelevant content [5, 6]. Continuous Integration (CI) pipelines, for example, may generate thousands of lines of build and test output, much of it boilerplate or repeated status updates [7]–[9]. Feeding such raw data directly into LLMs inflates inference cost and latency, wastes token budgets, and introduces noise that obscures the underlying signal. These inefficiencies reduce accuracy and usability while amplifying environmental costs [10]–[14].

Prior work has explored log compression, structural parsing, and deduplication, producing gains in storage and indexing efficiency [15]–[18]. Recent efforts even apply LLMs for anomaly detection and log filtering [19, 20]. Yet most techniques remain *task-agnostic*: they reduce size without regard for the semantic needs of downstream tasks, often preserving large volumes of irrelevant content that limit effectiveness in token-sensitive workflows.

We argue for a new paradigm: **task-aware text reduction pipelines** as a first-class component of language–data integration. Rather than indiscriminately compressing or parsing, these pipelines act as intelligent preprocessing layers that prioritize semantic relevance and explicitly treat the token budget of an LLM as an *attention budget*. By foregrounding task relevance, such pipelines promise three benefits: (1) scalability through token efficiency, (2) sustainability by lowering energy and infrastructure costs, and (3) accuracy by focusing attention on the most relevant signals. This paradigm is complementary to model- and architecture-level efficiency work [21]–[23] and retrieval-augmented generation and indexing approaches [24]–[26].

Contributions. This paper makes the following contributions:

- **Conceptual reframing:** We introduce input reduction as an *attention allocation problem*, positioning task-aware reduction as a complementary layer alongside model- and architecture-level efficiency methods [21]–[23].

- **Research agenda:** We outline open challenges in evaluation benchmarks, adaptive reduction strategies, token-budget–aware indexing, and sustainability metrics [24]–[26], charting a roadmap for scalable and environmentally responsible integration of LLMs with real-world data systems.

- **Domain generality:** We highlight opportunities for task-aware reduction beyond software logs, including healthcare [27]–[29], the Internet of Things [30, 31], and other data-intensive domains where verbosity threatens efficiency and accuracy.

Paper Organization. The rest of this paper is organized as follows. Section II provides background and reviews related work on log analysis and the use of LLMs in this context. Section III introduces our vision for task-aware text reduction pipelines as a new abstraction layer for language–data systems. Section IV outlines our research agenda and discusses key open challenges. Finally, Section V concludes the paper and suggests directions for future work.

II. BACKGROUND AND RELATED WORK

Prior work on log analysis has explored compression, structural parsing, and deduplication as strategies to manage the scale of text-rich data. For example, compression techniques improve log storage efficiency [15], while template-based methods such as LogZip [16], Drain [17], and Spell [18] reduce structural variability and support downstream processing. Earlier efforts in anomaly detection mined console logs to identify large-scale system problems [3]. These approaches have proven effective for storage and indexing but remain largely task-agnostic, often retaining significant amounts of irrelevant content.

Recent advances have begun to apply LLMs directly to log data. Qi et al. proposed LogGPT for anomaly detection, showing the potential of transformer-based models for log understanding [19]. More recently, Huang et al. introduced LoFI, a prompt-based method for extracting fault-indicating information from logs [20]. While promising, these approaches incur high inference costs or focus on specific tasks, rather than providing a general, reusable reduction layer. In contrast, we position *task-aware text reduction* as a general-purpose paradigm for directing scarce attention resources toward semantically relevant tokens.

A parallel line of work in natural language processing has focused on reducing the computational footprint of large models. Surveys on efficient Transformers [21] and architectural optimizations such as FlashAttention [22] propose model-level techniques to accelerate inference, while speculative decoding improves token generation efficiency [23]. These methods operate primarily at the level of model architecture, complementary to our focus on upstream input reduction as an attention-allocation problem.

Software engineering and sustainability research further highlights the environmental costs of large-scale computation [10]–[14, 32, 33]. These studies argue for approaches that balance performance with energy efficiency and ecological impact, motivating our exploration of reduction-oriented preprocessing as a sustainability principle.

In database and information retrieval research, retrieval-augmented generation (RAG) grounds LLMs in structured knowledge sources [24], and recent work explores efficient indexing strategies tailored for LLM workloads [25]. Classic IR theory on indexing and query optimization [26] provides additional motivation for treating task-aware preprocessing pipelines as first-class components in the language–data stack.

III. VISION: TASK-AWARE TEXT REDUCTION PIPELINE

We envision **task-aware text reduction pipelines** as a new abstraction layer for language–data systems. These pipelines operate between raw data streams and LLM inference, filtering and restructuring content so that only semantically relevant information is retained. Unlike compression or syntactic parsing [16]–[18], which aim to improve storage or indexing efficiency, task-aware pipelines are explicitly guided by the needs of downstream tasks such as failure triage, anomaly detection, or query answering.

This paradigm promises three key benefits. First, *scalability*: by reducing token counts before inference, pipelines lower latency and computational overhead. Second, *sustainability*: trimming unnecessary content reduces the carbon footprint of LLM-driven analysis by minimizing redundant computation and data transfer [11]–[13, 32, 33]. Third, *accuracy*: by exposing models only to semantically relevant signals, task-aware reduction can improve the precision and reliability of downstream outputs. In short, we treat the token budget of an LLM as an *attention budget*, and argue that reduction pipelines are essential to aligning scarce attention resources with the signals that matter most.

We propose three design principles for building such pipelines. **Task relevance first**: retain the information that contributes directly to the diagnostic or analytic objective, while aggressively filtering boilerplate or low-information content. **Token-budget awareness**: treat inference cost as a resource allocation problem, preserving the tokens that matter most under explicit budget constraints. **Hybrid structural–semantic reduction**: combine structural cues (e.g., templates, schema metadata, system events) with semantic methods (e.g., embeddings, task-specific prompts) to identify the segments most relevant to the task.

The scope of task-aware reduction extends well beyond Continuous Integration logs. Similar challenges arise in *cloud observability*, where vast telemetry streams contain only sparse anomaly signals; in *system monitoring*, where traces and events are verbose yet repetitive; and in *knowledge graphs*, where rich metadata and contextual annotations often overwhelm reasoning pipelines. Beyond software systems, *healthcare data* provides a striking example: clinical notes and electronic health records are lengthy and often dominated by boilerplate, yet only a subset of the content is relevant for clinical decision-making. Recent work on clinical summarization demonstrates how reduction can preserve diagnostic fidelity while eliminating unnecessary text [27]–[29]. Likewise, in the *Internet of Things*, continuous sensor and telemetry streams generate massive volumes of largely repetitive data, where the challenge lies in surfacing sparse and semantically important anomalies. Surveys and early frameworks highlight the opportunity for reduction and filtering as prerequisites for effective LLM-based reasoning in IoT domains [30, 31].

Across these settings, task-aware pipelines balance efficiency with fidelity, creating a unifying paradigm across domains from logs to medical notes to sensor data.

In short, task-aware text reduction offers a foundation for rethinking the role of preprocessing in language–data systems. By foregrounding relevance, these pipelines complement model-level efficiency advances and retrieval-based methods, establishing a new paradigm for scalable, sustainable, and accurate LLM–data integration.

IV. RESEARCH AGENDA AND OPEN CHALLENGES

Task-aware text reduction pipelines open a rich set of research opportunities at the intersection of natural language processing, databases, and software engineering. Realizing the full potential of this paradigm requires addressing several open challenges.

A. Automated Relevance Labeling

Manual annotation of task-relevant content is not scalable across large datasets or domains. Similar scalability challenges were observed in empirical analyses of Continuous Integration (CI) build data [9], which examined the interplay between build durations and breakages across thousands of integration runs and highlighted the difficulty of managing large, noisy datasets. In the context of root cause analysis, annotation is usually performed using manual log analyses to identify patterns associated with different types of errors [34, 35]. To address scalability challenges, future work should explore automated approaches to relevance labeling, including (1) heuristic rules that capture common signals such as error codes or exceptions, (2) weak supervision that combines noisy labels from multiple sources, and (3) LLM-assisted annotation to bootstrap relevance classifiers with minimal human effort [19, 20]. Recent work on root-cause analysis with LLM-based agents also suggests that domain-specific relevance signals can be learned and reused across settings [1, 2]. Such approaches would enable pipelines that automatically direct scarce attention budgets to the most meaningful tokens.

B. Adaptive Reduction Strategies

The optimal degree of reduction varies by context: compilation errors, for instance, may tolerate aggressive pruning, whereas sparse telemetry requires more conservative filtering. Designing adaptive pipelines that tailor reduction dynamically by failure type, domain, or query intent is an important direction for ensuring both efficiency and diagnostic fidelity. This may require hybrid approaches that combine learned models with domain-specific heuristics [1, 2] and should be evaluated alongside model-level efficiency methods such as efficient Transformers [21], FlashAttention [22], and speculative decoding [23]. Together, these methods would enable both computation-aware and attention-aware reduction.

C. Integration with Database and IR Systems

Task-aware reduction should not operate in isolation but integrate with existing data management infrastructures. One opportunity is *token-budget–aware indexing*, where reduced representations become first-class citizens in query engines. Another is hybrid retrieval pipelines that combine traditional IR methods with LLMs, leveraging reduction to shrink the search space and improve response latency. Embedding reduction into query planning itself may unlock new forms of task-aware optimization [24]–[26]. In each case, the goal is to align attention budgets with the segments most relevant to the query.

D. Sustainability Metrics and Benchmarks

A key motivation for reduction is sustainability, yet little work quantifies its real-world benefits. Future efforts should develop benchmarks that measure energy consumption, carbon footprint, and cost savings across LLM pipelines with and without reduction. These metrics would help compare techniques, guide system design, and motivate the adoption of reduction as an environmental as well as technical best practice [10]–[12, 32, 33]. Such benchmarks would clarify the sustainability impact of directing attention away from redundant tokens.

E. Beyond Logs

Although software logs highlight the challenge of verbosity, task-aware reduction applies equally to other domains.

In *telemetry and observability*, pipelines could surface sparse anomaly signals from large monitoring streams. In *system monitoring*, task-aware filters could trim redundant events while preserving causal signals. In *knowledge graphs*, reduction may help preprocess verbose metadata and contextual annotations for efficient reasoning.

Beyond software engineering, *healthcare data* offers a striking case. Clinical notes and electronic health records (EHRs) are lengthy, redundant, and often dominated by boilerplate, yet only a small subset of the content is critical for clinical decision-making. Recent advances in clinical summarization demonstrate that reduction can preserve essential signals while improving diagnostic support and mitigating hallucination risks [27]–[29].

Similarly, in the *Internet of Things*, continuous sensor and telemetry streams generate massive volumes of repetitive data, with only rare anomalies or outliers being relevant. Applying task-aware reduction here would enable LLMs to reason effectively over IoT data without being overwhelmed by redundancy. Surveys and prototypes highlight the need for scalable preprocessing and filtering in IoT-LLM integration, pointing to reduction as a prerequisite for real-world deployment [30, 31].

Each of these domains poses unique challenges, from domain-specific semantics in clinical text to high-frequency noise in IoT data. Yet all share the same fundamental need: directing limited attention budgets toward the tokens that matter most.

F. Call to the Community

We call on the community to treat task-aware reduction as a *foundational design principle* for language–data systems. Just as indexing and query optimization transformed relational databases, relevance-driven reduction has the potential to reshape how unstructured and semi-structured data is processed

in LLM pipelines. Achieving this vision requires collaboration across natural language processing, databases, and software engineering communities, building on both model-level efficiency advances [21, 22] and data-level retrieval frameworks [24, 25].

V. CONCLUSION

This paper has argued for **task-aware text reduction pipelines** as a cornerstone of future language–data systems. By foregrounding semantic relevance, these pipelines address a critical gap in current approaches, enabling scalable, accurate, and sustainable integration of LLMs with noisy, text-rich data sources [10]–[14, 32, 33].

We have positioned input reduction not as a storage or compression problem, but as an *attention allocation problem*: treating the token budget of an LLM as an attention budget that must be directed toward the most relevant signals. This shift in perspective opens a research agenda around automated relevance labeling [19, 20], adaptive reduction strategies [1, 2], hybrid structural–semantic techniques, and integration with database and retrieval systems [24]–[26].

The challenges we highlight are not confined to software logs. They extend to healthcare, where clinical notes and electronic health records demand task-aware summarization to support decision-making [27]–[29], and to the Internet of Things, where continuous sensor streams require filtering to surface rare anomalies [30, 31]. These diverse domains underscore the generality of reduction as a unifying paradigm for noisy, high-volume, text-rich data.

These directions complement model- and architecture-level efficiency methods [21]–[23], together defining a multi-layered approach to efficiency. We encourage the community to treat task-aware reduction not as an afterthought, but as a *foundational design principle*—just as indexing transformed relational databases, reduction has the potential to reshape how unstructured and semi-structured data is processed in LLM pipelines. Embracing this principle is a step toward building hybrid, scalable, and sustainable systems that define the next generation of language–data fusion.

REFERENCES

[1] D. Roy, X. Zhang, R. Bhave, C. Bansal, P. Las-Casas, R. Fonseca, and S. Rajmohan, "Exploring LLM-based agents for root cause analysis," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, 2024, pp. 208–219.

[2] Z. Wang, Z. Liu, Y. Zhang, A. Zhong, J. Wang, F. Yin, L. Fan, L. Wu, and Q. Wen, "RCAgent: Cloud root cause analysis by autonomous agents with tool-augmented large language models," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 4966–4974.

[3] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles*, 2009, pp. 117–132.

[4] J. Harty, H. Zhang, L. Wei, L. Pascarella, M. Aniche, and W. Shang, "Logging practices with mobile analytics: An empirical study on firebase," in *2021 IEEE/ACM 8th International Conference on Mobile Software Engineering and Systems (MobileSoft)*. IEEE, 2021, pp. 56–60.

[5] S. He, P. He, Z. Chen, T. Yang, Y. Su, and M. R. Lyu, "A survey on automated log analysis for reliability engineering," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–37, 2021.

[6] S. Gholamian and P. A. Ward, "A comprehensive survey of logging in software: From logging statements automation to log mining and analysis," *arXiv preprint arXiv:2110.12489*, 2021.

[7] T. Ghaleb, O. Abduljalil, and S. Hassan, "CI/CD configuration practices in open-source Android apps: An empirical study," *ACM Transactions on Software Engineering and Methodology*, 2024.

[8] F. Moriconi, T. Durieux, J.-R. Falleri, R. Troncy, and A. Francillon, "GHALogs: Large-scale dataset of GitHub Actions runs," in *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*. IEEE, 2025, pp. 669–673.

[9] T. A. Ghaleb, S. Hassan, and Y. Zou, "Studying the interplay between the durations and breakages of continuous integration builds," *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 2476–2497, 2023.

[10] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai *et al.*, "Sustainable AI: Environmental implications, challenges and opportunities," *Proceedings of machine learning and systems*, vol. 4, pp. 795–813, 2022.

[11] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[12] C. König, D. J. Lang, and I. Schaefer, "Sustainable software engineering: Concepts, challenges, and vision," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 5, pp. 1–28, 2025.

[13] S. Naumann, D. Schmidt, M. Dick, J. Kern, and J. M. Müller, "The GREENSOFT model: A reference model for green and sustainable software and its engineering," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 4, pp. 294–304, 2011.

[14] B. Penzenstadler, V. Bauer, C. C. Calero, and X. Franch, "Sustainability in software engineering: A systematic literature review," in *Proceedings of the 16th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2012.

[15] K. Yao, M. Sayagh, W. Shang, and A. E. Hassan, "Improving state-of-the-art compression techniques for log management tools," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 2748–2760, 2021.

[16] J. Liu, J. Zhu, S. He, P. He, Z. Zheng, and M. R. Lyu, "Logzip: Extracting hidden structures via iterative clustering for log compression," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2019, pp. 863–873.

[17] P. He, J. Zhu, Z. He, J. Li, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *2017 IEEE International Conference on Web Services (ICWS)*. IEEE, 2017, pp. 33–40.

[18] M. Du and F. Li, "Spell: Streaming parsing of system event logs," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 859–864.

[19] J. Qi, S. Huang, Z. Luan, S. Yang, C. J. Fung, H. Yang, D. Qian, J. Shang, Z. Xiao, and Z. Wu, "LogGPT: Exploring ChatGPT for log-based anomaly detection," in *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 2023, pp. 273–280.

[20] J. Huang, Z. Jiang, J. Liu, Y. Huo, J. Gu, Z. Chen, C. Feng, H. Dong, Z. Yang, and M. R. Lyu, "Demystifying and extracting fault-indicating information from logs for failure diagnosis," in *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2024, pp. 511–522.

[21] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, no. 6, Dec. 2022. [Online]. Available: https://doi.org/10.1145/3530811

[22] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," *Advances in neural information processing systems*, vol. 35, pp. 16344–16359, 2022.

[23] B. Spector and C. Re, "Accelerating LLM inference with staged speculative decoding," *arXiv preprint arXiv:2308.04623*, 2023.

[24] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[25] W. Chen, T. Bai, J. Su, J. Luan, W. Liu, and C. Shi, "KG-Retriever: Efficient knowledge indexing for retrieval-augmented large language models," May 2025, arXiv:2412.05547 [cs]. [Online]. Available: http://arxiv.org/abs/2412.05547

[26] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, ser. McGraw-Hill Computer Science Series. New York, NY: McGraw-Hill, 1983.

[27] H. O. Boll, A. O. Boll, L. P. Boll, A. A. Hanna, and I. Calixto, "DistillNote: LLM-based clinical note summaries improve heart failure diagnosis," *arXiv preprint arXiv:2506.16777*, 2025.

[28] G. Mehenni and A. Zouaq, "Ontology-constrained generation of domain-specific clinical summaries," in *Knowledge Engineering and Knowledge Management*, M. Alam, M. Rospocher, M. van Erp, L. Hollink, and G. A. Gesese, Eds. Cham: Springer Nature Switzerland, 2025, pp. 382–398.

[29] J. D. Oliveira, H. D. Santos, A. H. D. Ulbrich, J. C. Couto, M. Arocha, J. Santos, M. M. Costa, D. Faccio, F. O. Tabalipa, and R. F. Nogueira, "Development and evaluation of a clinical note summarization system using large language models," *Communications Medicine*, vol. 5, no. 1, p. 376, 2025.

[30] T. An, Y. Zhou, H. Zou, and J. Yang, "IoT-LLM: Enhancing real-world IoT task reasoning with large language models," *arXiv preprint arXiv:2410.02429*, 2024.

[31] F. Sarhaddi, N. T. Nguyen, A. Zuniga, P. Hui, S. Tarkoma, H. Flores, and P. Nurmi, "LLMs and IoT: A comprehensive survey on large language models and the internet of things," *Authorea Preprints*, 2025.

[32] "Measuring the energy consumption and efficiency of deep neural networks: An empirical analysis and design recommendations."

[33] E. Rolf, B. Packer, A. Beutel, and F. Diaz, "Striving for data-model efficiency: Identifying data externalities on group performance," in *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*, 2022. [Online]. Available: https://openreview.net/forum?id=_h_ikjOEGL_

[34] C. E. Brandt, A. Panichella, A. Zaidman, and M. Beller, "Logchunks: A data set for build log analysis," in *Proceedings of the 17th International Conference on Mining Software Repositories*, 2020, pp. 583–587.

[35] T. A. Ghaleb, D. A. Da Costa, Y. Zou, and A. E. Hassan, "Studying the impact of noises in build breakage data," *IEEE Transactions on Software Engineering*, vol. 47, no. 9, pp. 1998–2011, 2019.