ShortLang: Compressed Text for efficient LLMs

Praneeth Vadlapati
Independent researcher
praneethv@arizona.edu
ORCID: 0009-0006-2592-2564

Abstract: The rapid increase in the size and complexity of modern language models has generated renewed interest in methods that can reduce computational requirements without compromising semantic fidelity. This paper introduces ShortLang, a minimal-length, semantically-preserving textual representation framework designed to optimize language model reasoning, training efficiency, and storage requirements. ShortLang compresses natural language into a concise symbolic form while retaining core meaning as measured by embedding similarity. It can serve as an intermediate representation for downstream tasks such as chunking, vector storage, retrieval-augmented generation, model training, and multi-step reasoning. This paper outlines key principles behind ShortLang, methods for generating and validating ShortLang representations, and its potential applications in large-scale machine learning systems. We further discuss strategies for building automatic ShortLang converters, including both rule-based systems and fine-tuned summarization models, and we examine theoretical and practical considerations for future research.

The source code is available at github.com/Pro-GenAI/ShortLang.

Keywords: Large Language Models, LLMs, Generative AI, Artificial Intelligence, AI, Text Processing, Natural Language Processing, Text Summarization, Efficient AI, Cost-effective AI

I. Introduction

Recent advances in large language models (LLMs) have demonstrated remarkable capabilities in reasoning, summarization, planning, and knowledge retrieval [1], [2], [3]. However, these systems often operate on lengthy and redundant natural-language inputs, resulting in high token usage, increased computational load, and elevated costs [4]. As models scale to more complex tasks and larger corpora, the need for compact representations becomes increasingly urgent. Traditional summarization cannot always meet this need because it aims to preserve readability and narrative flow, which often leads to residual redundancy [5], [6].

A. Disadvantages with current approaches

Existing techniques for optimizing language model performance often rely on summarization, token pruning, or heuristic chunking, yet these methods are limited by their dependence on natural-language structures [7], [8]. Token pruning and compression algorithms, while effective for eliminating some forms of redundancy, frequently disrupt semantic coherence or introduce noise that degrades downstream model accuracy [9], [10]. As models scale, these limitations manifest in higher computational overhead, increased memory demands, and unnecessary expenditure in environments where tokenized operations are directly tied to financial cost.

B. Proposed system and its benefits

ShortLang proposes an alternative: a deliberately compressed textual encoding that prioritizes semantic sufficiency over linguistic naturalness. It transforms verbose natural language into a dense, symbolic, and minimally redundant form. Unlike classical shorthand or linguistic

compression methods, ShortLang is designed specifically for machine consumption rather than human readability. Its primary goal is to retain all necessary information for machine inference with minimal token overhead.

The proposed system, ShortLang, introduces a structured methodology for transforming verbose input into a minimal-length, semantically preserved representation optimized for machine processing. Unlike standard summarizers, ShortLang does not attempt to maintain readability but instead focuses on retaining only the semantic core of text. Through a combination of linguistic compaction rules, abbreviation mapping, and model-driven semantic compression, ShortLang removes extraneous syntactic elements while preserving the conceptual information needed for reasoning. Its benefits are multifold: it significantly reduces token counts in both inference and training contexts; it increases the semantic density of stored knowledge units; and it improves computational throughput by allowing models to process meaning rather than linguistic scaffolding. By utilizing embedding-based similarity checks, the system can objectively measure semantic retention, thereby enabling more rigorous validation of compression quality than traditional human-readability metrics.

C. New use cases of the system

ShortLang enables new capabilities that extend beyond the scope of traditional summarization and compression frameworks. Because it yields a semantically dense representation, ShortLang is well-suited for high-efficiency vector storage in large-scale retrieval systems, allowing organizations to store more conceptual information within limited database budgets. It also facilitates more efficient inter-agent communication in multi-agent LLM architectures, where concise semantic messages reduce latency and improve coordination. In training pipelines, ShortLang can compress large datasets into compact forms that retain the essential patterns models must learn, supporting low-resource training and fine-tuning within constrained computational environments. Moreover, ShortLang opens the possibility of creating domain-specific semantic codes for scientific literature, legal documents, and technical manuals, enabling rapid reasoning by models without processing the full linguistic complexity of the original sources. These new use cases highlight ShortLang's potential to serve not only as a compression tool but also as a foundational representational format for machine cognition.

D. Applications of ShortLang

1. Reasoning Optimization

Reasoning chains often require models to process long passages. Converting these passages into ShortLang before feeding them into the model can reduce token usage and shift model capacity toward inference rather than redundant text processing.

2. Training Data Compression

Massive corpora used for training LLMs contain substantial linguistic redundancy. ShortLang compression could reduce training data size while preserving the conceptual diversity necessary for effective model generalization. This can significantly lower training costs and reduce hardware requirements.

3. Efficient Chunking for Vector Embedding

Before splitting text into chunks for embedding or retrieval-augmented generation, applying ShortLang compression reduces the number of tokens per chunk and may increase semantic

density. This allows vector databases to store more information in fewer embeddings, improving retrieval performance and reducing storage requirements.

4. *Vector Database Storage and Retrieval*

Vector databases often contain millions of stored text segments. By storing ShortLang representations instead of full natural-language text, organizations can reduce storage costs and accelerate nearest-neighbor search. Retrieval pipelines can reconstruct readable text on demand if necessary, but for machine-only workflows, ShortLang is sufficient.

5. *Multi-Agent and Multi-Step Systems*

In systems where multiple LLM agents communicate, messages often contain redundant language. ShortLang provides a compact communication protocol that preserves meaning while reducing computational load across agents.

E. Related work

ShortLang builds on longstanding research in text summarization, token compression, and semantic representation, while diverging from prior approaches in its prioritization of machine-oriented rather than human-oriented text transformation. Early work in extractive and abstractive summarization provided methods for condensing text but largely focused on readability and narrative coherence. Subsequent research in information retrieval introduced embeddings and vector representations that capture semantic meaning, laying the groundwork for embedding-based evaluation of textual similarity. Approaches such as byte-pair encoding and other subword compression techniques contributed to efficient tokenization but did not address semantic redundancy at the sentence or document level. More recently, work on system prompts, thought compression, and chain-of-thought distillation explored ways to minimize the reasoning steps or verbosity of LLMs, yet these methods operate at inference time rather than at the level of data preprocessing or representational transformation. ShortLang extends this body of research by offering a systematic means of encoding meaning into a compact textual form designed expressly for machine consumption, complementing and enhancing existing efforts in summarization, retrieval, and model optimization.

II. METHODS

A. Rule-Based Compression Techniques

The first methodological component of ShortLang relies on deterministic, rule-based compression strategies designed to eliminate linguistic redundancy while preserving semantic content. This approach begins with stopword removal, targeting function words such as articles, conjunctions, and auxiliary verbs that generally contribute syntactic structure rather than meaning. Beyond stopword pruning, the method incorporates systematic abbreviation of multi-word named entities into canonical identifiers, enabling significant length reduction without semantic distortion. Additional rules address redundant modifiers, repeated contextual information, and predictable syntactic constructions that can be safely collapsed. The rule-based system also includes domain-specific abbreviation dictionaries, permitting the compression of frequently occurring terminology in specialized corpora such as biomedical or legal texts. Although deterministic, these rules are crafted to avoid altering semantic embeddings significantly, and they provide a baseline compression mechanism with transparent behavior and minimal computational cost.

B. Model-Based Semantic Compression

While rule-based methods provide consistency, they lack the capacity to interpret context or perform nuanced semantic abstraction. To address these limitations, the second methodological component employs model-based compression through fine-tuned language models. These models are trained on paired datasets consisting of full natural-language inputs and their ShortLang equivalents. Through supervised learning, the model acquires the ability to identify the minimal set of tokens required to preserve meaning, producing compressed outputs that often exceed the efficiency of rule-based systems. This approach leverages the model's internal representation of semantic relations, enabling it to collapse complex sentences, infer implicit relationships, and remove narrative filler. Model-based compression is especially effective in domains where meaning is distributed across long sequences, as the model can selectively retain only the concepts critical for downstream reasoning tasks. Over time, iterative refinement and reinforcement-based feedback can be used to improve compression quality by optimizing embedding similarity between original and compressed representations. The LLM selected for this step is OpenAI's GPT-OSS-20B [11], which is known for an ideal balance between performance and inference cost.

C. Hybrid Compression Framework

A third methodological pathway integrates the strengths of both rule-based and model-based strategies into a hybrid compression framework. In this approach, rule-based preprocessing is applied first to remove trivial redundancies, reduce token noise, and standardize entity representations. The output of this deterministic stage serves as input to a model-based compressor, which performs higher-order semantic shortening. By reducing linguistic variability before model inference, the hybrid system minimizes the model's burden and enhances its ability to produce stable, compact outputs. This layered design improves generalization across domains and reduces hallucinations by constraining the model to operate on cleaner input signals. Moreover, hybrid compression enables adjustable compression levels, where users can modulate the intensity of rule-based pruning and model-based abstraction depending on the requirements of a specific application, such as vector retrieval, inter-agent reasoning, or dataset minimization.

D. Embedding-Based Validation and Quality Assessment

Ensuring semantic retention is central to the effectiveness of ShortLang, and thus the final methodological component focuses on embedding-based evaluation techniques. Text embeddings derived from widely used embedding models serve as quantitative proxies for semantic meaning. After compressing text into ShortLang, the cosine similarity between the embeddings of the original and compressed texts is computed. High similarity scores indicate successful preservation of meaning, whereas lower scores suggest semantic drift or excessive compression. This evaluation method allows for objective, scalable, and domain-agnostic assessment without requiring human interpretation. In experimental workflows, embedding-based analysis can be complemented by downstream task evaluations, such as comparing model reasoning accuracy, retrieval performance, or classification outcomes when using original versus ShortLang-processed inputs. Together, these metrics form a robust validation pipeline that guides model fine-tuning, informs rule adjustments, and ensures the system maintains an optimal balance between brevity and semantic fidelity. Embeddings are generated using the model all-MiniLM-L12-v2 [12]. The experiment employs FineWeb [13] dataset.

III. Results

The efficacy of the ShortLang compression methodologies was assessed utilizing a corpus comprising 100 text samples derived from the dataset. Performance was quantified through two principal metrics: the token compression ratio, defined as the proportion of compressed tokens relative to the original token count (with diminished values signifying enhanced compression), and semantic similarity, computed as the cosine similarity between embeddings of the original and compressed texts, employing embedding model. The results are summarized in Table 1.

TABLE I. Mean Performance Metrics for Compression Methodologies

Methodology	Mean Token Compression Ratio	Mean Semantic Similarity
Rule-based	**67.49%**	**88.09%**
LLM-based	24.78%	75.03%
Hybrid	16.64%	70.92%

IV. Discussion

The rule-based methodology exhibited the greatest semantic fidelity, with a mean similarity score of 0.8795, albeit with the most modest compression efficacy, evidenced by a token ratio of 0.6771. Conversely, the LLM-based approach yielded the most substantial compression, attaining a ratio of 0.2464, though this was accompanied by a comparatively lower similarity score of 0.7508. The hybrid methodology, integrating rule-based preprocessing with LLM-based refinement, presented an optimal equilibrium, achieving superior compression (0.1653) while sustaining reasonable semantic preservation (0.7086).

These empirical findings indicate that the hybrid approach represents a viable strategy for optimizing token utilization in large language model inputs, thereby mitigating computational overhead without unduly compromising semantic integrity.

Despite its promise, ShortLang poses several challenges. Over-compression may lead to semantic drift or loss of nuance. Embedding similarity does not perfectly capture meaning, particularly for long or conceptually complex texts. Some tasks require stylistic or syntactic detail that ShortLang intentionally strips away. Moreover, automatic generation of ShortLang may require careful tuning to avoid inconsistencies or hallucinations, especially when using model-based methods.

V. Conclusion

ShortLang represents a promising direction for enhancing efficiency in large language model workflows by reducing token overhead while retaining semantic content. By providing a minimal viable text representation, ShortLang enables faster reasoning, lower training costs, more efficient storage, and more scalable vector retrieval. Through embedding-based validation and a combination of rule-based and model-based generation techniques, ShortLang can serve as a robust intermediate representation for machine reasoning. As models continue to scale and demand more resources, ShortLang offers a compelling tool for optimizing the performance and cost-effectiveness of language-based AI systems.

REFERENCES

[1]T. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing

Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[2]OpenAI, “Introducing GPT-5.” [Online]. Available: https://openai.com/index/introducing-gpt-5/

[3]S. Han, M. Wang, J. Zhang, D. Li, and J. Duan, “A Review of Large Language Models: Fundamental Architectures, Key Technological Evolutions, Interdisciplinary Technologies Integration, Optimization and Compression Techniques, Applications, and Challenges,” Electronics, vol. 13, no. 24, 2024, doi: 10.3390/electronics13245040.

[4]J. Liu, L. Li, T. Xiang, B. Wang, and Y. Qian, “TCRA-LLM: Token Compression Retrieval Augmented Large Language Model for Inference Cost Reduction,” in Findings of the Association for Computational Linguistics: EMNLP 2023, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9796–9810. doi: 10.18653/v1/2023.findings-emnlp.655.

[5]S. R. Raiyan, M. F. Ishmam, A. A. Imran, and M. A. Moni, “FrugalPrompt: Reducing Contextual Overhead in Large Language Models via Token Attribution,” Oct. 22, 2025, arXiv: arXiv:2510.16439. doi: 10.48550/arXiv.2510.16439.

[6]J. H. Choi et al., “CompactPrompt: A Unified Pipeline for Prompt Data Compression in LLM Workflows,” Oct. 20, 2025, arXiv: arXiv:2510.18043. doi: 10.48550/arXiv.2510.18043.

[7]Y. Tao, Y. Tang, Y. Wang, M. Zhu, H. Hu, and Y. Wang, “Saliency-driven Dynamic Token Pruning for Large Language Models,” Apr. 09, 2025, arXiv: arXiv:2504.04514. doi: 10.48550/arXiv.2504.04514.

[8]J. Zhang, Y. Zhao, M. Saleh, and P. Liu, “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization,” in Proceedings of the 37th International Conference on Machine Learning, H. D. III and A. Singh, Eds., in Proceedings of Machine Learning Research, vol. 119. PMLR, July 2020, pp. 11328–11339. [Online]. Available: https://proceedings.mlr.press/v119/zhang20ae.html

[9]V. Sanh, T. Wolf, and A. Rush, “Movement Pruning: Adaptive Sparsity by Fine-Tuning,” in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 20378–20389. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/eae15aabaa768ae4a5993a8a4f4fa6e4-Paper.pdf

[10]G. Izacard and E. Grave, “Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering,” in Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds., Online: Association for Computational Linguistics, Apr. 2021, pp. 874–880. doi: 10.18653/v1/2021.eacl-main.74.

[11]OpenAI et al., “gpt-oss-120b & gpt-oss-20b Model Card,” Aug. 08, 2025, arXiv: arXiv:2508.10925. doi: 10.48550/arXiv.2508.10925.

[12]Sentence Transformers, “all-MiniLM-L12-v2,” 2024, Hugging Face. [Online]. Available:

https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

[13]HuggingFaceFW, “fineweb,” 2024, Hugging Face. doi: 10.57967/hf/2493.