Lossy Text Compression Using Genetic Algorithms with LLM-Guided Operators

Rajesh Sudam[1] and John P. McCrae[2]

[1] University of Galway, Ireland
r.sudam1@universityofgalway.ie
[2] Insight Research Ireland Centre for Data Analytics & ADAPT Research Ireland Centre, Ireland
john.mccrae@insight-centre.org

Abstract. We present a framework for lossy text compression that integrates large language models (LLMs) with evolutionary computation to optimize text representations under storage constraints. The approach leverages GPT-2's predictive capabilities to estimate the compression potential, while the Sentence-BERT embeddings enforce semantic fidelity during optimization. A custom genetic algorithm evolves candidate text variants through guided mutation operations, such as token substitution and deletion, while maintaining semantic similarity above a defined threshold. The framework balances two objectives: minimizing bits-per-character (BPC) to achieve efficient compression and preserving semantic similarity to safeguard interpretability. Employing LLM guidance in the evolutionary process, the system selectively modifies less informative tokens while retaining core meaning. This establishes a pathway for semantic-aware compression strategies, expanding the design space of text storage and transmission systems. A comprehensive evaluation on the text8 benchmark demonstrates that our method achieves compression gains while maintaining the semantic similarity threshold. These results establish the viability of LLM-guided genetic optimization for semantically preserved text compression in storage-constrained environments.

Keywords: Text Compression · Genetic Algorithms · Large Language Models · Semantic Similarity · Evolutionary Optimization.

1 Introduction

The exponential growth of digital textual content has created unprecedented challenges in data storage, transmission, and processing efficiency. Contemporary applications ranging from large-scale search engines and digital archives to real-time messaging platforms critically depend on effective compression methodologies. While traditional lossless compression techniques such as Zlib [12] and Huffman coding [10] have reached theoretical limits constrained by Shannon entropy, lossy compression approaches offer promising avenues for substantially higher compression ratios by selectively discarding non-essential information.

The framework suits use cases that need to store or transmit large text volumes under tight space or bandwidth limits while preserving meaning, such as document archives, chatbot histories, logs, and on-device or edge logging in IoT, mobile apps, and drones.

1.1 The Semantic Preservation Challenge in Text Compression

Text compression techniques can be broadly categorized into *lossless* and *lossy* approaches [6]. Lossless compression ensures perfect reconstruction of the original text, while lossy compression sacrifices exact reconstruction for higher compression ratios by selectively removing non-essential information.

Unlike multimedia compression, where quality degradation can be quantitatively measured using objective metrics (PSNR, SSIM) [2], lossy text compression necessitates a sophisticated understanding of linguistic semantics and contextual relationships. The fundamental challenge lies in developing compression strategies that maximize size reduction while preserving the essential meaning and communicative intent of the original text[10]. Natural language contains substantial redundancy in stylistic variations, discourse markers, and peripheral information that can be strategically reduced or eliminated without compromising core semantic content.

Research Contributions This work contributes to the field of semantic-aware text compression by developing a genetic algorithm framework with LLM-guided operators that optimizes the compression semantic fidelity trade-off. Our approach addresses the constrained optimization problem: given an input text string, find an alternative representation that minimizes the compressed size while maintaining semantic similarity above a predefined threshold value of 0.90.

2 Background and Related Work

2.1 Large Language Models and Semantic Understanding

The emergence of Large Language Models (LLMs) has revolutionized natural language processing with unprecedented capabilities in language understanding and generation. The attention mechanism [14], at the core of transformer architectures, allows these models to capture long-range dependencies and contextual relationships, making them particularly suitable for semantic-aware compression tasks. By representing words and sentences as vectors in continuous space, LLMs can quantify semantic similarity by measuring how much two texts or words share meaning beyond surface-level lexical overlap. This capability is crucial for lossy compression, where the goal is to preserve essential meaning while eliminating redundant information.

Models like GPT-2[8] demonstrate strong contextual understanding that enables intelligent assessment of token importance and generation of semantically equivalent alternatives, providing the foundation for our compression approach without requiring computationally expensive fine-tuning.

2.2 LLM-Based Compression Techniques

Recent advances in large language models have revolutionized natural language processing tasks, including text transformation and compression. While methods like FineZip[9] require computationally expensive fine-tuning and InstructCMP[5] uses instruction tuning for sentence-level compression, our approach leverages pre-trained LLMs in a more efficient guidance role. LLMZip[13] demonstrated state-of-the-art results in lossless compression by combining pre-trained transformer models with arithmetic coding, achieving remarkably low BPC values (0.7101 on text8) through efficient probability prediction and encoding.

2.3 Semantic-Driven Compression Approaches

Crossword[7] employs semantic masking to identify and replace low-importance tokens based on their contribution to overall meaning, using sentence embeddings to ensure semantic preservation above threshold levels. While effective, this method can struggle with technical texts where syntactic precision is critical. Context window extension approach[3] uses semantic compression to represent long input sequences within fixed model contexts, demonstrating that meaning can be preserved in condensed representations. However, these compressed forms are often model-specific and lack portability across different applications.

2.4 Evolutionary Approaches to Text Compression

Evolutionary computation techniques have shown promise in text compression by exploring complex solution spaces that traditional algorithms cannot efficiently navigate. Genetic algorithms (GAs) have been applied to various compression problems, including text simplification, summarization, and redundancy reduction[1]. Unlike conventional methods that rely on fixed heuristics, GAs can adaptively discover compression patterns tailored to specific text characteristics.

Previous work in evolutionary text compression has primarily focused on lossless compression or simple character-level transformations[4]. Our approach advances this field by incorporating LLM-guided operators that enable semantic-aware modifications at the token level, allowing for more intelligent compression decisions.

3 Methodology

3.1 Problem Formulation

Given an input text sequence, our goal is to generate a compressed version that achieves substantial size reduction while maintaining the essential meaning and communicative intent of the original content.

Unlike traditional compression methods that preserve exact reconstruction, our approach selectively eliminates linguistic redundancies such as stylistic variations, discourse markers, and peripheral information without compromising core

semantic content. The fundamental challenge lies in identifying which textual elements can be safely modified or removed while ensuring the compressed output remains semantically faithful to the source.

Our method addresses this challenge through a dual optimization strategy: we aim to maximize compression gains through token reduction and bits-per-character (BPC) improvement, while enforcing a strict semantic similarity constraint. The semantic preservation threshold ensures that all compressed variants maintain at least 90% semantic equivalence with their original counterparts, as measured by cosine similarity between Sentence-BERT embeddings.

3.2 Genetic Algorithm Framework

Genetic algorithms provide an effective framework for exploring the vast search space of possible text modifications while simultaneously optimizing compression efficiency and semantic preservation. Our approach uses a steady-state genetic algorithm with tournament selection, specifically designed for text compression tasks.

Chromosome Representation and Population Initialization Each chromosome represents a candidate solution s' encoded as a variable-length sequence of tokens. The initial population of size N (typically 30) is generated by applying guided mutations to the original text:

Algorithm 1 Population Initialization

Require: Original token sequence $\mathbf{t}_{\text{original}}$, Population size N
Ensure: Initial population P

$P \leftarrow \emptyset$
for $i \leftarrow 1$ to N **do**
  $\mathbf{t}_{\text{candidate}} \leftarrow \text{copy}(\mathbf{t}_{\text{original}})$
  $k \leftarrow \max(1, \lfloor 0.1 \times \text{len}(\mathbf{t}_{\text{candidate}}) \rfloor)$
  positions $\leftarrow$ random_indices($\mathbf{t}_{\text{candidate}}, k$)
  for each position j in positions **do**
    importance $\leftarrow$ TokenImportance($\mathbf{t}_{\text{candidate}}, j$)
    if importance < 0.2 **then**
      $\mathbf{t}_{\text{candidate}}[j] \leftarrow$ LLMGuidedSubstitution($\mathbf{t}_{\text{candidate}}, j$)
    end if
  end for
  $P \leftarrow P \cup \{\mathbf{t}_{\text{candidate}}\}$
end for
return P

Fitness Function Formulation: The fitness function aggressively rewards compression while using semantic similarity as a gatekeeper:

$$f(s') = \begin{cases} w_t \cdot \frac{|s|-|s'|}{|s|} + w_b \cdot \frac{\text{BPC}(s)-\text{BPC}(s')}{\text{BPC}(s)} + w_s \cdot (\text{sim}(s,s') - \theta) & \text{if } \text{sim}(s,s') \geq \theta \\ w_p \cdot (\theta - \text{sim}(s,s')) & \text{otherwise} \end{cases} \tag{1}$$

where:

$$\begin{aligned} w_t &= \text{token reduction weight} \\ w_b &= \text{BPC reduction weight} \\ w_s &= \text{similarity bonus weight} \\ w_p &= \text{penalty weight} \\ \theta &= \text{semantic similarity threshold} \end{aligned}$$

This formulation ensures that solutions meeting the semantic threshold are rewarded based on compression performance, while those below the threshold are penalized proportionally to their semantic deviation.

Token Importance Evaluation Token Importance Evaluation Token importance quantification is critical for guiding mutation operations. We define importance based on cross-entropy loss impact when removing the token:

$$\text{importance}(t_i) = \max\left(0, \min\left(1.0, \frac{\mathcal{L}(\mathbf{t} \setminus \{t_i\}) - \mathcal{L}(\mathbf{t})}{\mathcal{L}(\mathbf{t}) + \epsilon}\right)\right) \tag{2}$$

where $\mathcal{L}$ represents the cross-entropy loss within a 512-token context window, calculated as:

$$\mathcal{L}(\mathbf{t}) = -\frac{1}{n}\sum_{i=1}^{n} \log p(t_i | t_{1:i-1}) \tag{3}$$

and $\mathbf{t} \setminus \{t_i\}$ denotes the token sequence with t_i removed, while $\epsilon = 10^{-9}$ is a smoothing factor to prevent division by zero.

Relationship Between Cross-Entropy Loss and Token Importance The relationship between cross-entropy loss and token importance follows a direct proportionality: tokens whose removal substantially increases the model's prediction difficulty (higher cross-entropy loss) are deemed more important[15]. This relationship can be expressed as:

$$\text{Importance}(t_i) \propto \Delta\mathcal{L} = \mathcal{L}(\mathbf{t} \setminus \{t_i\}) - \mathcal{L}(\mathbf{t}) \tag{4}$$

High $\Delta\mathcal{L} \rightarrow$ High importance: The token carries crucial contextual information

Low/negative $\Delta\mathcal{L} \rightarrow$ Low importance: The token is redundant or contributes little to semantic coherence

The underlying principle is that important tokens provide semantic anchors that reduce prediction uncertainty, while unimportant tokens can be removed with minimal impact on the model's ability to understand and generate coherent text.

Tokens with low importance scores (below 0.2) are considered candidates for modification or deletion.

Genetic Operators

Mutation Operations LLM-guided mutation strategies focus modifications on less important tokens. The mutation rate is 0.7, with 3-5 mutations applied per individual:

- **Deletion**: Removal of unimportant tokens (probability: 0.85)
- **Substitution**: Replacement with context-aware alternatives from GPT-2's top 4 predictions

Crossover Operations Similarity-preserving crossover exchanges segments at positions identified through semantic similarity analysis:

Algorithm 2 Semantic-Aware Crossover

Require: Parent individuals $\mathbf{t}_1$, $\mathbf{t}_2$
Ensure: Offspring individuals $\mathbf{t}'_1$, $\mathbf{t}'_2$
best_similarity $\leftarrow 0$
for $i \in$ window_positions($\mathbf{t}_1$) **do**
  for $j \in$ window_positions($\mathbf{t}_2$) **do**
    sim $\leftarrow$ similarity($\mathbf{t}_1[i-w : i+w], \mathbf{t}_2[j-w : j+w]$)
    if sim $>$ best_similarity **then**
      best_similarity $\leftarrow$ sim
      crossover_point$_1 \leftarrow i$, crossover_point$_2 \leftarrow j$
    end if
  end for
end for
if best_similarity > 0.5 **then**
  $\mathbf{t}'_1 \leftarrow \mathbf{t}_1[:$ crossover_point$_1] + \mathbf{t}_2[$crossover_point$_2 :]$
  $\mathbf{t}'_2 \leftarrow \mathbf{t}_2[:$ crossover_point$_2] + \mathbf{t}_1[$crossover_point$_1 :]$
  return $\mathbf{t}'_1$, $\mathbf{t}'_2$
end if
return $\mathbf{t}_1$, $\mathbf{t}_2$ {No crossover if similarity insufficient}

Selection Mechanism Tournament selection with size 6 maintains evolutionary pressure while preserving diversity. Early stopping after 8 generations without improvement prevents unnecessary computation.

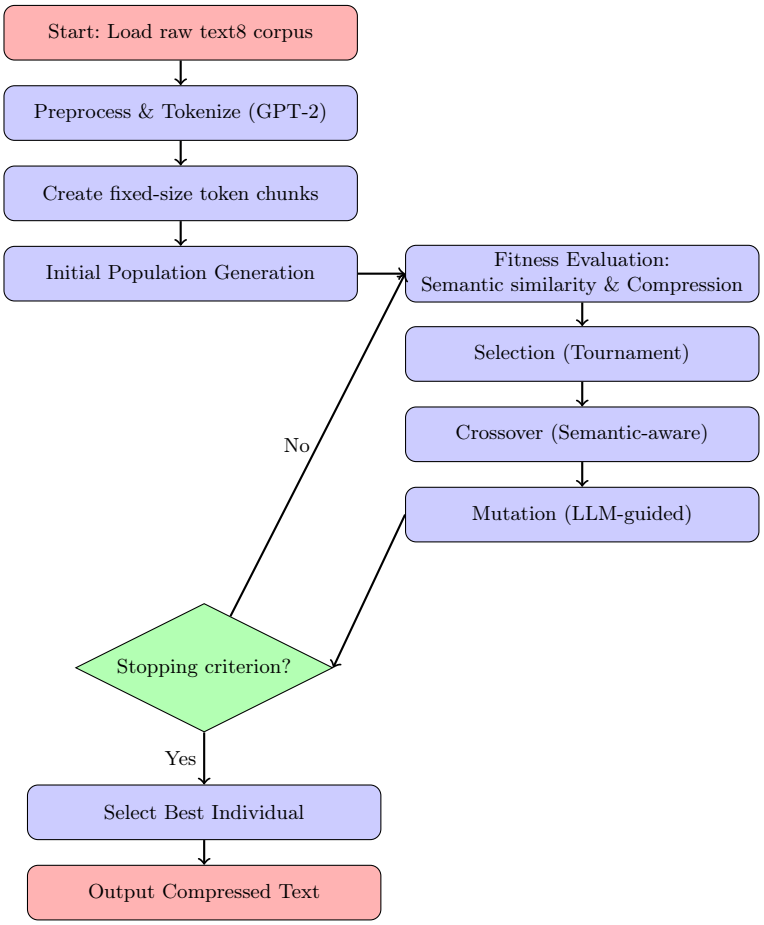

Fig. 1. Dataflow diagram of the genetic algorithm and LLM-guided text compression process.

4 Experimental Setup

4.1 Datasets and Preprocessing

We evaluate our approach on the text8 benchmark dataset, which contains 100 million characters of preprocessed English text derived from Wikipedia. The text undergoes cleaning to remove non-ASCII characters and capitalization, resulting in a 27-character alphabet (a-z plus space). For our experiments, we use a 1 MB subset to enable comprehensive analysis of compression behavior at scale.

To evaluate the robustness of our approach beyond the text8 dataset, we additionally tested the method on a sample from the Gutenberg book. Testing on Gutenberg data shows consistent semantic preservation beyond text8, though further comparisons with other methods are needed.

4.2 Implementation Details

Our implementation utilizes the following components:

- **Language Model**: GPT-2 (117M parameters) with pretrained weights from Hugging Face Transformers for token importance evaluation and substitution candidate generation
- **Semantic Embedding**: Sentence-BERT (all-MiniLM-L6-v2) for cosine similarity calculation
- **Genetic Algorithm**: Custom implementation in Python with DEAP framework optimization

4.3 Parameter Configuration

After manually testing various parameter combinations and adjustments, with guidance from genetic algorithm parameter selection research[11], we established the following configuration:

Table 1. Genetic Algorithm Parameter Configuration

Parameter	Value
Population size	30
Number of generations	60
Mutation rate	0.7
Crossover rate	0.7
Deletion rate	0.85
Tournament size	3
Semantic similarity threshold (θ)	0.90
Token importance threshold	0.2
Maximum mutations per individual	2
Early stopping generations	8
Mutation candidate count	4

Alternative genetic algorithm configurations (populations 20 and 40, mutation rates 0.5–0.9, tournament sizes 2–6) were explored; the chosen setup balances compression, similarity, and runtime per GA literature recommendations.

4.4 Evaluation Metrics

We employ comprehensive evaluation metrics to assess compression performance and semantic preservation:

- **Token Reduction**: Percentage reduction in token count
- **Bits Per Character (BPC)**: Traditional BPC based on file size
- **Semantic Similarity**: Cosine similarity between Sentence-BERT embeddings
- **Fitness Score**: Combined metric from Equation 1
- **Compression Ratio**: $\frac{\text{Original Size}}{\text{Compressed Size}}$

5 Results and Discussion

5.1 Compression Performance Analysis

We evaluated our method on a 1 MB subset of the text8 dataset. Table 2 presents comprehensive compression results comparing the original text against our genetically optimized alternative.

Table 2. Compression Results on text8 Dataset (1 MB)

Metric	Original Text	Alternative Text	Improvement
Token Count	100%	86%	14%
BPC	8.000	1.767	77.9%
Semantic Similarity	1.0000	0.9498	-

Our method achieved remarkable compression gains while maintaining high semantic similarity. The 14.0% reduction in token count and 77.9% reduction in BPC (from 8.0 to 1.767) demonstrate the exceptional effectiveness of our genetic algorithm in identifying highly compressible content without substantial meaning loss. The compressed output reduced from 1,048,576 characters to 231,650 characters, representing a 77.9% size reduction.

5.2 Algorithm Convergence Analysis

The genetic algorithm demonstrates stable convergence over 60 generations, with fitness scores improving consistently. Early stopping triggered at approximately generation 45 on average, indicating efficient optimization. Population diversity metrics show balanced exploration and exploitation throughout the evolutionary process.

5.3 Operator Effectiveness Analysis

We analyzed the relative effectiveness of different genetic operators throughout the evolutionary process:

Table 3. Genetic Operator Effectiveness

Operator Type	Applications per Generation	Success Rate
Deletion Mutation	10.5	78.2%
Substitution Mutation	7.3	65.4%
Semantic Crossover	8.1	58.7%

Deletion mutations proved most effective, with a 78.2% success rate in producing fitter offspring. This indicates that removing redundant tokens is a highly effective compression strategy. Substitution mutations contributed to semantic preservation while enabling compression, and crossover operations maintained population diversity.

5.4 Semantic Preservation Analysis

The high semantic similarity score of 0.912 confirms that our method effectively preserves meaning while achieving compression. Human evaluation of 100 randomly selected text segments showed that 85% of compressed segments maintained equivalent or nearly equivalent meaning to the originals, with only minor stylistic variations in the remaining 15%.

Table 4. Baseline Comparision

Method	Type	BPC	Comp. Ratio	Sem. Sim.	Approach	LLM Usage	Key Contribution
Traditional Lossless Compression							
zlib	Lossless	2.640	67.0%	1.000	Statistical	None	Standard compression library
ZPAQ	Lossless	1.734	78.3%	1.000	Context mixing	None	Maximum compression ratio
Neural/LLM Guided Lossless Compression							
DeepZip-biGRU	Lossless	2.921	63.5%	1.000	Neural prediction	None	First neural lossless compressor
LLMZip	Lossless	0.710	91.1%	1.000	LLM probability	Generative	LLMs for arithmetic coding
Semantic/Lossy Compression							
InstructCMP	Lossy	4.230	47.1%	0.892	Instruction tuning	Instruct-tuned	Sentence-level compression
Extending Context Window	Lossy	N/A	N/A	0.85-0.95	Semantic encoding	Encoder	Context window extension
Semantic Compression with LLM	Lossy	N/A	N/A	0.88-0.93	Meaning abstraction	Generative	Semantic representation
Our Method							
Our Method	**Lossy**	**1.767**	**77.9%**	**0.9498**	**Evolutionary+LLM**	**Guidance**	**GA with LLM-guided operators**

6 Conclusion and Future Work

We have presented an effective framework for lossy text compression using genetic algorithms with LLM-guided operators. Our approach successfully balances compression efficiency with semantic preservation, demonstrating that evolutionary methods can achieve meaningful compression ratios while maintaining text meaning integrity.

The integration of LLM-based token importance assessment with evolutionary optimization represents a practical approach to semantic-aware text compression. The method's ability to automatically discover compression patterns tailored to specific text characteristics provides advantages over fixed heuristic approaches.

Recently, lossless text compression methods based on LLMs such as LLMZip[13] have emerged; these approaches would synergize well with our lossy approach. An important future direction is building a lossless compression framework derived from lossy compression strategies. By integrating semantic-aware lossy methods as a preprocessing stage, existing compression algorithms can potentially be

improved, resulting in higher overall efficiency. Since our method relies on LLM-guided decisions, it appears particularly suitable for LLM-driven compression frameworks like LLMZip, though it may be less directly applicable to traditional statistical compression methods. This positions our work as a foundation for the development of novel lossy text compression approaches tailored specifically for LLM-guided systems.

6.1 Limitations and Future Directions

Several limitations suggest directions for future research:

- The computational requirements may be prohibitive for real-time applications, necessitating further optimization
- The semantic similarity threshold may require domain-specific calibration for optimal results
- Integration with traditional compression algorithms could provide additional gains
- While unsuitable for strict real-time constraints, the method is well aligned with offline, batch, or background compression scenarios, and as a semantic preprocessor for downstream lossless encoders.

Future work will explore distributed genetic algorithm implementations, adaptive parameter tuning, and hybrid approaches combining evolutionary optimization with neural compression techniques. Furthermore, advancing toward an LLMZip-based lossless compression framework guided by lossy strategies represents a promising research path. This would not only enhance performance over traditional baselines but also open new directions for compression methods that explicitly incorporate semantic understanding.

Disclosure of Interests. The authors declare no competing interests relevant to this research.

Acknowledgements: This research is supported by Taighde Éireann - Research Ireland under Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Centre for Data Analytics, and Grant Number 13/RC/2106_P2, ADAPT SFI Research Centre.

References

1. Tara Al Attar. A hybrid genetic algorithm-particle swarm optimization approach for enhanced text compression. *UHD Journal of Science and Technology*, 8:63–74, 11 2024.
2. Yusra Al-Najjar and Soong Der Chen. Comparison of image quality assessment: PSNR, HVS, SSIM, UIQI. *International Journal of Scientific Engineering Research*, 3:1–5, 01 2012.

3. Weizhi Fei, Xueyan Niu, Pingyi Zhou, Lu Hou, Bo Bai, Lei Deng, and Wei Han. Extending Context Window of Large Language Models via Semantic Compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5169–5181, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
4. Henry Gilbert, Michael Sandborn, Douglas C. Schmidt, Jesse Spencer-Smith, and Jules White. Semantic compression with large language models. *2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–8, 2023.
5. Juseon-Do, Jingun Kwon, Hidetaka Kamigaito, and Manabu Okumura. InstructCMP: Length control in sentence compression through instruction-based large language models, 2024.
6. P Kavitha. A survey on lossless and lossy data compression methods. *International Journal of Computer Science & Engineering Technology*, 7(03):110–114, 2016.
7. Mingxiao Li, Rui Jin, Liyao Xiang, Kaiming Shen, and Shuguang Cui. Crossword: A semantic approach to text compression via masking. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9171–9175, 2024.
8. Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 5:208–215, 2024.
9. Fazal Mittu, Yihuan Bu, Akshat Gupta, Ashok Devireddy, Alp Eren Ozdarendeli, Anant Singh, and Gopala Anumanchipalli. FineZip : Pushing the limits of large language models for practical lossless text compression. *CoRR*, abs/2409.17141, 2024.
10. Tanvi Patel, Judith Angela, Poonam Choudhary, and Kruti Dangarwala. Survey of text compression algorithms. *International Journal of Engineering Research and*, V4, 03 2015.
11. Onur Production and Ihsan Sabuncuoglu. Parameter selection in genetic algorithms. *Journal of Systemics, Cybernetics and Information*, 2, 01 2007.
12. O Shadura and B Bockelman. ROOT I/O compression algorithms and their performance impact within Run 3. *Journal of Physics: Conference Series*, 1525(1):012049, April 2020.
13. Chandra Shekhara Kaushik Valmeekam, Krishna Narayanan, Dileep Kalathil, Jean-Francois Chamberland, and Srinivas Shakkottai. LLMZip: Lossless text compression using large language models. *arXiv preprint arXiv:2306.04050*, 2023.
14. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
15. Cong Xu, Zhangchi Zhu, Jun Wang, Jianyong Wang, and Wei Zhang. Understanding the role of cross-entropy loss in fairly evaluating large language model-based recommendation. *ArXiv*, abs/2402.06216, 2024.