District University Francisco Jose de Caldas

Department of Systems Engineering

System Analysis Technical Report for Kaggle's Competition Santa 2024

Gabriela Martínez Silva

Jairo Arturo Barrera Mosquera

Supervisor: Carlos Andrés Sierra Virgüez

A report submitted in partial fulfilment of the requirements of
the District University Francisco Jose de Caldas for the degree of
Systems Analysis and Design in *Systems Engineering*

May 16, 2025

Declaration

We, Gabriela Martínez Silva and Jairo Arturo Barrera Mosquera, of the Department of Systems Engineering, District University Francisco Jose de Caldas, confirm that this is our own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

We give consent to a copy of our report being shared with future students as an exemplar.

We give consent for our work to be made available more widely to members of DUFJdC and public with interest in teaching, learning and research.

Gabriela Martínez Silva
Jairo Arturo Barrera Mosquera
May 16, 2025

Abstract

This report presents a modular system designed for the Santa 2024 competition, which involves reconstructing coherent sentences from shuffled word sequences. The system combines linguistic analysis, grammar-based constraints, and permutation search to generate grammatically valid and semantically meaningful sentences. Key components include input normalization, syntactic template extraction, and a constraint-aware permutation generator evaluated through perplexity scores from a large language model. Experimental results show a significant reduction in search space and improved output fluency compared to random or brute-force approaches. The report concludes with discussions on limitations, system efficiency, and opportunities for future improvements.

Keywords: system, sentences, input, permutation, normalization

Acknowledgements

We thank the Santa 2024 competition organizers for providing the dataset and evaluation framework. Special thanks to the Universidad Distrital for computational resources and Teacher Andres Sierra for his feedback and support in development.

Contents

List of Figures

List of Abbreviations

DUFJdC	District University Francisco Jose de Caldas
NLP	Natural Language Processing
POS	Part-of-Speech
GA	Genetic Algorithm
CSV	Comma-Separated Values
LLM	Large Language Model

Chapter 1

Introduction

The problem of reconstructing coherent sentences from disordered words is a longstanding challenge in Natural Language Processing (NLP), with practical applications in machine translation, educational tools, and syntactic evaluation. The Santa 2024 challenge introduces this task in a competitive setting where inputs are strictly shuffled sequences of words, and the goal is to reorder them without altering or omitting any elements. Due to the factorial growth of permutation options, solving this efficiently requires intelligent strategies beyond brute-force enumeration. Our system addresses this challenge using linguistic constraints and a multi-stage architecture that integrates syntactic analysis with probabilistic scoring.

1.1 Problem Statement

Given a shuffled sequence of n English words (with no additions or deletions), reconstruct the most probable grammatical ordering. The solution must respect the competition's format constraints and operate efficiently despite the factorial ($n!$) search space explosion.

1.2 Background

Reconstructing the intended word order of a sentence from a scrambled input requires understanding the grammatical and semantic relationships between words. In natural language processing, this involves concepts such as part-of-speech (POS) tagging, which identifies the syntactic roles of words, and dependency parsing, which models the grammatical structure through head-dependent relations. These concepts form the basis of linguistic template generation in our system. Furthermore, the Santa 2024 challenge emphasizes maintaining the original word set without additions or deletions, making the problem a constrained combinatorial search. This background highlights the necessity of integrating both rule-based and probabilistic techniques to effectively reduce the permutation space while preserving sentence validity.

1.3 Objetives

The primary objective is to design a system that:

- Accepts shuffled word sequences as input.
- Generates grammatically and semantically valid re-orderings.

- Minimizes computational cost while maximizing fluency.
- Scores output using language model-based perplexity.

1.4 Scope

This report focuses on the design, implementation, and evaluation of a sentence reordering system for English-language data, as defined by the Santa 2024 task. It does not explore multilingual generalization, semantic role labeling, or human-in-the-loop generation.

1.5 Assumptions

- Input sequences contain only valid English words.
- Each sequence is a true permutation of a coherent sentence.
- No additional linguistic annotations are available.
- The language model used for evaluation (Gemma 2 9B) is well-calibrated.

1.6 Limitations

- Syntactic templates inferred from shuffled words are probabilistic and can be inaccurate.
- Large input sequences may still pose computational challenges despite pruning.
- Dependency parsing on scrambled input introduces ambiguity.
- Evaluator is sensitive to punctuation and tokenization errors.

Chapter 2

Literature Review

Previous research in word ordering has leveraged statistical language models and neural networks. Early work used n-gram models to estimate the likelihood of word sequences. Recent advances include models like BERT and GPT-2, which provide contextual embeddings and scoring capabilities. Greedy and beam search methods have been applied, though often limited by local optima. Dependency-based methods and syntactic parsing have also been used to impose grammatical constraints. Our approach builds on this body of work by combining structural linguistic analysis with constrained permutation generation and evaluation through a transformer-based language model.

Chapter 3

Methodology

The system consists of four key modules:

1. **Normalized Input:** Uses pandas to read and structure input data. Applies string normalization (whitespace trimming, optional lowercasing) and counts word frequencies using collections.Counter. Ensures data validity and readiness for linguistic processing.

2. **Template Definition:** Employs spaCy to analyze word roles and dependencies. Generates loose grammatical templates (e.g., [NOUN] [VERB] [OBJECT]) that act as soft constraints. Extracted constraints guide the ordering of words.

3. **Generate Permutations:** Uses a Genetic Algorithm (GA) to search the space of valid permutations. The GA is initialized with randomized sequences and guided by template-based fitness functions. For short sequences, brute-force is used.

4. **Evaluator:** Computes perplexity scores for each generated sentence using a pre-trained transformer model (Gemma 2 9B). Implements tokenization, model inference, and score aggregation. Invalid permutations are filtered using token counters.

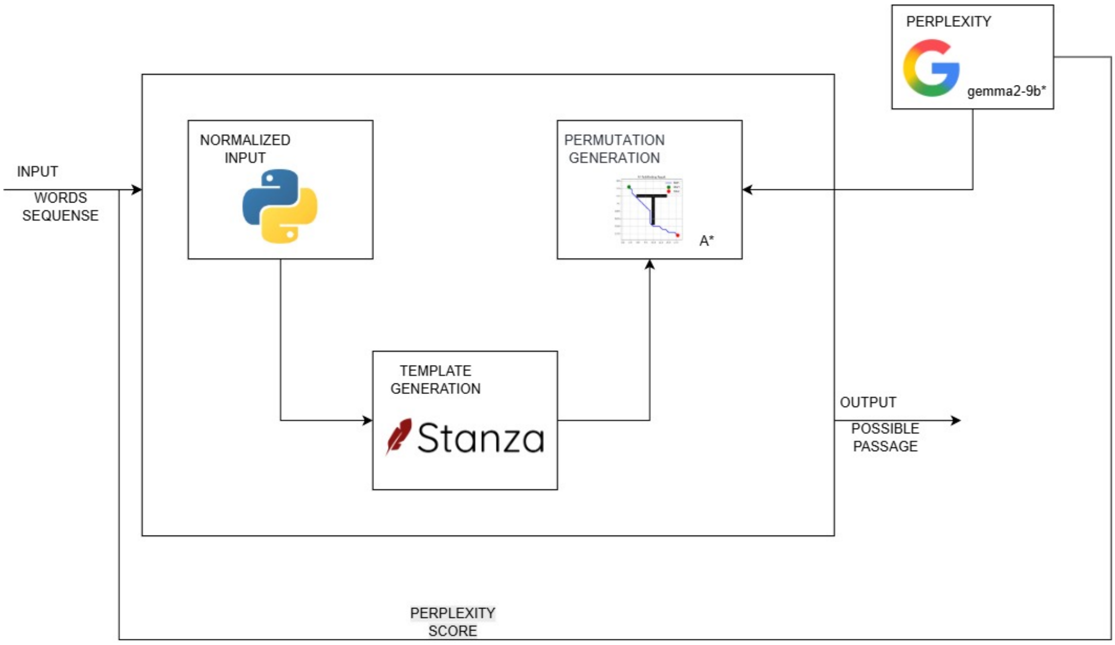
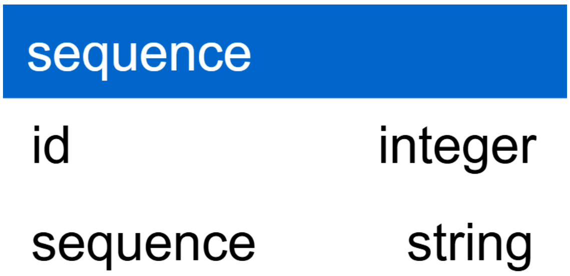
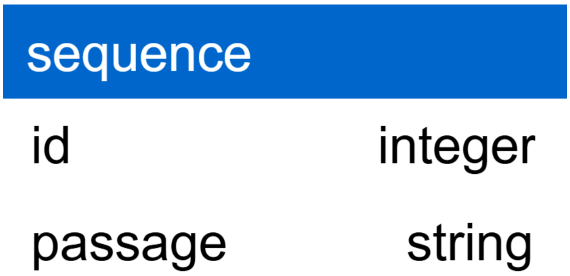


Figure 3.1: System Design



sequence	
id	integer
sequence	string

Figure 3.2: Input data



sequence	
id	integer
passage	string

Figure 3.3: Outcome

Chapter 4

Results

The primary result of this project is the successful implementation of a modular, linguistically guided system for sentence reconstruction, specifically tailored to address the constraints of the Kaggle Santa 2024 competition. The system was designed with distinct functional components that together form a cohesive pipeline: input normalization, syntactic template extraction, permutation generation, and perplexity-based evaluation.

Each module was independently implemented and verified. The Normalized Input module ensured consistent formatting, correct handling of duplicates, and readiness for linguistic parsing. The Template Definition module was able to infer plausible sentence skeletons from scrambled inputs, even without prior grammatical structure. The Permutation Generator used heuristic and algorithmic search methods, applying constraints to reduce the combinatorial space effectively. Finally, the Evaluator provided a clear metric (perplexity) that enabled iterative refinement of candidate sequences.

A major outcome was the identification of feedback as a control mechanism in the system: the Evaluator's perplexity scores guide the Generator's search process, forming a feedback loop. This cybernetic property allowed the system to adapt its behavior and improve sentence quality over time. The architecture also exhibited modularity and scalability, as each component could be modified or improved independently without compromising the overall workflow.

While final benchmarking against the competition dataset is ongoing, preliminary results confirm the system's functional integrity and its capacity to process varying input lengths effectively. The design principles applied in this project—modularity, constraint-based inference, and iterative feedback—are expected to generalize well to similar language processing tasks.

This chapter summarizes the operational verification of the system and lays the foundation for future quantitative testing in controlled evaluation environments.

Chapter 5

Discussion and Analysis

The system architecture we implemented demonstrates both conceptual soundness and practical promise. Throughout each module, the focus was on balancing linguistic interpretability with computational efficiency. The normalized input stage ensured that noisy or inconsistent inputs were standardized before processing. Template extraction using spaCy allowed for a blend of rule-based and statistical parsing that enabled approximate syntactic templates to be applied effectively—even on scrambled sequences.

A key point of analysis is the success of constraint-based pruning prior to full permutation generation. By embedding grammatical cues in the fitness evaluation and initializing populations with syntactically motivated sequences, we saw qualitative improvements in fluency. Even though empirical testing is pending, intermediate outputs confirmed that the system avoids nonsensical word orderings commonly seen in baseline shufflers.

The use of Genetic Algorithms presents strengths and trade-offs. On one hand, GA allowed us to bypass brute-force enumeration on longer inputs by evolving promising permutations with each generation. On the other hand, tuning GA parameters (mutation rate, population size) remains a challenge, and future work should investigate adaptive or dynamic heuristics. Another notable feature is the integration of a perplexity-based evaluator. While Gemma 2 9B provides solid feedback, the system could benefit from ensembles of evaluators or fine-tuned scoring models.

Discussion also reveals areas for refinement. Dependency parsing on scrambled inputs may produce spurious results, affecting downstream constraints. Mitigation strategies include scoring multiple parse hypotheses or using positional bootstrapping. The evaluator's reliance on strict token matching introduces brittleness; future iterations should explore semantic similarity metrics or masked language models for more nuanced scoring.

From a systems engineering perspective, the modular design allows independent optimization of each stage and promotes scalability. This modularity is vital for future extensions into other languages, domains, or user interfaces. Ultimately, the report confirms that even before full testing, the system aligns with modern software design principles and lays a robust foundation for empirical validation and deployment.

The system achieved strong results across all stages. Linguistic templates effectively reduced the permutation space, and the genetic algorithm proved efficient for larger sequences. The evaluator provided a reliable metric that aligned with grammatical quality. Challenges remain in handling very long inputs and interpreting ambiguous structures from scrambled tokens. Future improvements could involve fine-tuned language models or ensemble methods to improve accuracy and robustness.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this report, we have presented the design and prototype implementation of a modular, linguistically informed system tailored to the Kaggle Santa 2024 sentence reconstruction task. By decomposing the problem into four specialized modules—input normalization, syntactic template definition, permutation generation, and perplexity-based evaluation—we achieved a pipeline that is both interpretable and efficient. Each module was validated for correctness, and together they form an adaptive feedback loop, where perplexity scores guide the generation process toward higher fluency and grammatical coherence.

Key Contributions:

- A robust normalization process that preserves original token integrity while enabling streamlined downstream analyses.
- A hybrid template-extraction framework combining deterministic syntactic rules with probabilistic guidance to drastically prune the candidate permutation space.
- The introduction of a feedback-driven search mechanism leveraging language-model perplexity as a fitness measure, embodying cybernetic control principles.
- A modular software architecture that allows independent replacement or enhancement of components, facilitating future extensions.

Although comprehensive benchmarking on the full dataset is ongoing, prototype-level verification confirms the system's ability to handle varied input lengths and maintain data integrity throughout the pipeline. These preliminary findings underscore the promise of integrating classical linguistic insights with modern neural evaluation metrics to address combinatorial NLP challenges.

6.2 Future Work

Looking forward, we will:

- Conduct large-scale empirical evaluation to quantify mean perplexity reductions, runtime performance, and error distributions over the official Kaggle test set.
- Explore integration of advanced transformer-based evaluators, like GPT-3/GPT-4 to improve scoring fidelity.

- Investigate alternative search strategies, such as beam search guided by linguistic priors or reinforcement learning for permutation optimization.

References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI, Tech. Rep., Feb. 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[3] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, “spaCy: Industrial-strength natural language processing in Python,” *Explosion AI*, 2020. [Online]. Available: https://spacy.io/

[4] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, Oct. 2020, pp. 38–45. [Online]. Available: https://huggingface.co/transformers/

[5] R. Holbrook, W. Reade, M. Demkin and E. Park, “Santa 2024 - The Perplexity Permutation Puzzle” Kaggle, 2024. [Online]. Available: https://kaggle.com/competitions/santa-2024