Project Report: A Predictive Model for Jargon Identification and Simplification Date: August 26, 2025 Author: Gemini Project: Development of a Heuristic-Based Jargon Simplifier

1. Introduction The goal of this project was to develop a system capable of identifying and simplifying domain-specific jargon within a given sentence. The project evolved from a simple dictionary-lookup approach to a more sophisticated predictive model. This model learns the statistical and grammatical characteristics of jargon from a sample dataset and applies this knowledge to identify and simplify new, previously unseen jargon in novel sentences. This report details the data collection methods, the rationale for the chosen non-neural approach, and the key challenges and lessons learned throughout the development process.

2. Data Collection Method The project utilized a two-pronged data collection strategy to build its predictive capabilities. The model required not only examples of jargon but also a baseline understanding of general language use to statistically differentiate the two.

Primary Jargon Corpus (dataset_1.csv): The foundational data was a user-provided CSV file containing pairs of sentences. Each pair consisted of an "Original" sentence, where a jargon term was explicitly annotated with asterisks (e.g., **collusion**), and a "Simplified" sentence with a plain-language equivalent. This file served as our labeled training data. It provided the model with concrete examples of what constitutes jargon within a specific context (a mix of legal and medical terms). This dataset was used to build a "jargon frequency profile"—a statistical count of how often each word appears in a specialized, jargon-heavy context.

General English Baseline Corpus (NLTK's Brown Corpus): To teach the model what isn't jargon, a large, balanced corpus of general English was required. For this, we selected the Brown University Standard Corpus of Present-Day American English, accessible through the Natural Language Toolkit (NLTK). The Brown Corpus contains one million words of text from 15 different genres (news, fiction, academic, etc.), making it an excellent representation of standard language. By creating a frequency profile from this corpus, we established a statistical baseline against which the jargon corpus could be compared. The contrast between a word's frequency in the jargon corpus versus the general corpus became the primary signal for identifying potential jargon.

3. Non-Neural Approach Chosen and Rationale While modern NLP often defaults to neural network architectures, a heuristic-based statistical model was deliberately chosen for this project. This non-neural approach combines statistical analysis with rule-based grammatical heuristics to make its predictions. The model identifies jargon by evaluating two primary clues for each word in a new sentence:

Statistical Rarity Score: A "jargon score" is calculated for each word. This score is the ratio of the word's normalized frequency in the jargon corpus to its normalized frequency in the general English corpus. A high score indicates that the word is statistically much more likely to appear in a specialized context, making it a strong jargon candidate. This is a targeted application of the core concept behind TF-IDF.

Grammatical Role: Using NLTK's part-of-speech (POS) tagger, the model analyzes the grammatical structure of the sentence. A powerful heuristic was applied: jargon terms are predominantly nouns. The model was configured to only consider words tagged as nouns (NN, NNS, NNP, etc.) as potential jargon.

A word is ultimately flagged as jargon only if it surpasses a predefined rarity score and is identified as a noun.

Rationale for Choosing This Approach:

Interpretability and Transparency: Unlike the "black box" nature of many neural networks, every decision made by this model is fully explainable. We can precisely see a word's jargon score and its POS tag and understand why it was flagged. This makes debugging and tuning the model

straightforward.

Data Efficiency: Neural networks, particularly large language models, are data-hungry and require massive datasets to learn effectively. The provided dataset_1.csv is small. A statistical model is far more data-efficient and can derive meaningful patterns from a limited dataset without the significant risk of overfitting that a neural network would face.

Simplicity and Computational Speed: The model is computationally lightweight. The "training" phase simply involves counting word frequencies, which is extremely fast. The prediction (inference) phase is also rapid, making it suitable for real-time applications without requiring specialized hardware like GPUs.

Direct and Targeted Solution: The chosen approach directly models the intuitive definition of jargon: specialized nouns that are used infrequently in common parlance. It is a simple, elegant, and highly effective solution tailored specifically to the problem at hand.

4. Challenges and Lessons Learned The development process, though successful, presented several challenges that provided valuable insights.

Challenge 1: Evolving the Core Logic from Memorization to Prediction. The initial request could be interpreted as building a simple dictionary from the provided dataset and then performing a lookup. The key challenge was shifting the objective from merely simplifying known jargon to predicting unknown jargon.

Lesson Learned: This led to the crucial insight that the dataset should be treated not as a list to memorize, but as a statistical sample of a specialized domain. The real value was in comparing this sample to a general language baseline, which unlocked the model's predictive power.

Challenge 2: Environment Instability and Tooling. Throughout the interactive development, the execution environment encountered persistent package management and network errors. This prevented the successful installation and use of libraries like spaCy and at times even NLTK.

Lesson Learned: A stable, correctly configured development environment is paramount. The persistent errors forced a practical and flexible pivot from the more feature-rich spaCy library to the more lightweight and fundamental NLTK parser. This demonstrated the importance of adaptability and having alternative tools available to accomplish the same core task (syntactic parsing).

Challenge 3: The Nuance of "Simplification". The final step of the model relies on WordNet to provide a simpler synonym for the identified jargon. However, WordNet is a lexical database, not a context-aware engine.

Lesson Learned: Automatic synonym replacement is not a perfect science. WordNet may occasionally provide a synonym that doesn't perfectly fit the sentence's context or isn't demonstrably "simpler." This highlights that while jargon identification can be highly accurate, true context-aware simplification remains a more complex NLP challenge, often requiring more advanced generative models. Our model provides a robust "best-effort" substitution.

Challenge 4: Heuristic Tuning. The model's performance is sensitive to the JARGON_SCORE_THRESHOLD. A value of 100.0 was chosen as a starting point, but this is a tunable hyperparameter.

Lesson Learned: In a real-world application, this threshold would need to be carefully optimized using a validation dataset to balance precision (not flagging non-jargon) and recall (finding all the jargon). This underscores that even non-neural models have key parameters that govern their behavior and require empirical tuning for optimal performance.