Text Complexity Analyzer

Jan Chylarecki

November 28, 2025

Overview

The **Text Complexity Analyzer** is a Python-based tool designed to assess the linguistic complexity of a given text. The program performs tokenization, and lexical analysis. It computes the entropy and frequency-based measurements to produce a 0–100 complexity score.

The goal of the project is to gain practical experience in computational linguistics and text processing, as well as in Python programming. This aligns with my Data Science and Artificial Intelligence Bachelor's learning path.

Features

- Regex-based tokenization
- Word count and sentence count
- Average word length
- Average sentence length
- Length-normalized lexical diversity measure - *Guiraud's R*
- Entropy of the word frequency distribution
- Hapax legomena count
- Rare word ratio
- Weighted complexity score (0–100 scale)
- Word frequency histogram (Top N words)

Methodology

Given a text sample, let:

- N = total number of tokens (words)
- V = vocabulary size (number of unique tokens)
- f_i = frequency of each unique token

Guiraud's R

$$R = \frac{V}{\sqrt{N}}$$

This is a slightly better, robust and versatile alternative to the traditional type-token ratio (TTR). The TTR becomes unreliable when facing long texts, which can bias the data.

Lexical Entropy

$$H = -\sum_{i=1}^{V} p_i \log_2(p_i)$$

where:

$$p_i = \frac{f_i}{N}$$

Normalized entropy is computed as:

$$H_{\text{norm}} = \frac{H}{\log_2(V+1)}$$

Repeat Penalty

$$\text{Penalty} = 1 - \frac{\text{Hapax}}{V}$$

Average Word Length

$$awl = \frac{\sum \text{length(word)}}{N}$$

Average Sentence Length

$$asl = \frac{N}{S}$$

S being the number of sentences.

Complexity Score

Each component is normalized into $[0, 1]$ and combined with weights:

$$C_{\text{raw}} = 0.30 \cdot asl_{\text{norm}} + 0.25 \cdot H_{\text{norm}} + 0.20 \cdot R_{\text{norm}} + 0.15 \cdot \text{Penalty} + 0.10 \cdot awl_{\text{norm}}$$

The final score is scaled:

$$C = 100 \cdot C_{\text{raw}}$$

Usage

Run the program from a terminal:

```
python text_analyzer.py
```

Enter or paste the text when prompted. A histogram of the top N most frequent words will be displayed in a separate file.

Requirements

- Python 3.x
- matplotlib

Install dependencies:

```
pip install matplotlib
```

Future Work

- Support for additional languages and characters
- Improved sentence segmentation
- More readability formulas
- Command-line interface flags
- Web or GUI version

License

This project is distributed under the MIT License.