NLP Graphic Tool: Research Assignment Option 2

Natural Language Processing Course

Tesfay Hagos Weldegebriel (VR544201)
tesfayhagos.weldegebriel@studenti.univr.it

February 26, 2026

Course Information

- **University**: University of Verona
- **Professor**: Prof. Matteo Cristani
- **Subject**: Natural Language Processing

---

1 Introduction

This report documents the implementation of a **Graphic Tool for Document Processing**, developed as part of the Research-Oriented Assignment (Option 2) for the Natural Language Processing course. The tool is designed to provide a modern, interactive interface for common text analysis tasks, with specialized support for both **English** and **Tigrinya ()**.

2 System Architecture

The application is built using a modular Python architecture:

- **Core Pipeline (`nlp_pipeline.py`)**: Handles text cleaning, language detection, tokenization, lemmatization (English/Tigrinya), and relevance calculations.
- **Graphic UI (`app.py`)**: An interactive web interface built with **Streamlit**, featuring real-time analysis, visualizations with **Plotly**, and data export capabilities.

3 Requirement Implementation

The project successfully addresses all five requirements specified in the assignment:

3.1 2.1 Eliminate Stopwords

The tool removes common stopwords that carry little semantic weight.

- **English**: Uses the standard NLTK stopword corpus.
- **Tigrinya**: Integrates with the `tigrinya-nlp` library using a minimal stopword configuration.

3.2 2.2 Lemmatization

- **English**: Implements the `WordNetLemmatizer` with POS (Part-of-Speech) tagging for high accuracy.
- **Tigrinya**: Implements a custom **rule-based stemmer** developed with verification from a native speaker. It handles possessive, plural, and object pronoun suffixes (e.g., stripping -, -, -).

3.3 2.3 Frequency Computation

The pipeline generates a frequency table of all lemmas/stems after filtering. Results are displayed in interactive tables and visualized via bar charts and pie charts.

3.4 2.4 Distance from Strategic Points

Distances are measured relative to the total number of tokens:

1. **Distance from Start**: $\frac{Firstoccurrenceposition}{Totaltokens}$
2. **Distance from End**: $\frac{Totaltokens-Lastoccurrenceposition}{Totaltokens}$

3.5 2.5 Compound Relevance Indices

A custom relevance index is calculated following the 50% frequency + 50% earliness formula:

$$Relevance = 0.5 \times FrequencyScore + 0.5 \times EarlinessScore$$

Where:

- *Frequency Score* $= \frac{TermFrequency}{MaxFrequency}$
- *Earliness Score* $= 1 - \frac{AveragePosition}{TotalTokens}$

4 Key Features

- **Language Detection**: Automatic detection between English and Tigrinya.
- **Interactive Visualizations**: Scatter plots (Distance vs. Pos), Bar charts (Relevance), and Pie charts (Distribution).
- **Data Export**: Analysis results can be downloaded in **CSV** or **JSON** formats.
- **Modern UI**: Features a dark-themed glassmorphic design for enhanced user experience.

5 Conclusion

The NLP Graphic Tool provides a robust framework for document analysis, fulfilling academic requirements while pushing beyond basic processing by supporting a low-resource language like Tigrinya with specialized linguistic rules.