Week 1 – PRACTICAL

Introduction to NLP with Text Frequency Analysis

Introduction

This practical introduces you to the foundations of Natural Language Processing (NLP). You will explore a text corpus, perform word frequency analysis, and apply simple preprocessing steps. The goal is to understand how raw text can be converted into structured information, the initial stage of the NLP pipeline.

Tools Overview

- **NLTK (Natural Language Toolkit)**: A classical Python NLP library that provides access to corpora and basic NLP functions.
- **Matplotlib / Wordcloud**: For simple visualizations of text data.
- **Corpus:** A Large collection of texts stored in digital form, used for analysis in NLP or linguistics. A corpora is multiple text collections.

Part A – Exploring a Text Corpus

Q1). Load Shakespeare's Hamlet from the NLTK Gutenberg corpus and display the first 50 words.

Hints:

- Use `` `nltk.corpus.gutenberg.words("shakespeare-hamlet.txt")` ``.
- Print the length of the text to see how many tokens it contains.
- Display the first 50 tokens.

Q2). Count how often each word appears in Hamlet and print the 20 most common words.

Hints:

- Use `` `nltk.FreqDist(text)` ``.
- Call `` `.most_common(20)` `` to display results.

- Plot the top 20 frequencies with `.plot(20)`.

Part B – Preprocessing Raw Text

Q1). Convert all words to lowercase and remove punctuation tokens.

Hints:

- Use `.lower()` on each word.
- Keep only tokens that are alphabetic (`word.isalpha()`).

Q2) Remove English stopwords (e.g., the, and, of).

Hints:

- Import `stopwords` from `nltk.corpus`.
- Use a set of stopwords and filter them out of your word list.

Q3) Recalculate the top 20 most frequent words after preprocessing. Compare your results with the unprocessed version.

Part C – Relative Frequencies and Visualisation

Q1) Compute the relative frequency of words using:

$$f(w) = \frac{\mathrm{count}(w)}{\text{total words}}$$

Hints:

- Divide each word's count by the total number of tokens in your cleaned text.

Q2) Create a Wordcloud from the cleaned text to visualise the most frequent words.

Hints:

- Use the `` `WordCloud` `` class from the `` `wordcloud` `` library.
- Display with `matplotlib`.

Part D – Extension Task

Q1). Choose a second text (e.g., Austen's Emma) from the Gutenberg corpus. Apply the same preprocessing and frequency analysis.

Q2). Compare the top words between Hamlet and Emma. What differences do you notice in vocabulary and style?

Reflection Questions

- Why do stopwords dominate before preprocessing?
- How does relative frequency differ from raw counts?
- What insights and limitations do you notice when using word frequency analysis to understand a text?
- How does this exercise connect to the NLP pipeline discussed in the lecture?