

# CSE508 Winter 2024 A1 Report

Rahul Ajith — 2021083

09-02-2024

## Overview

This report details the development and outcomes of three interconnected systems designed to process and query a dataset of text files. The project is structured into three main tasks: preprocessing text files, creating an inverted index, and implementing a positional index for phrase queries. Each task employs specific methodologies, makes certain assumptions, and achieves distinct results, which are elaborated below.

## 0.1 Preprocessing Text Files

### Approach and Methodology

The preprocessing step involves reading text files from a dataset, converting the text to lowercase, tokenizing the text into words, removing stopwords and punctuation, and finally, saving the preprocessed text into new files. The process utilizes the NLTK library for tokenization and stopwords removal.

### Assumptions

- Text content is primarily in English.
- Punctuation and stopwords (commonly used words that do not contribute significantly to the semantic meaning) can be safely removed without losing critical information.
- Lowercasing all text standardizes the dataset, making future processing and querying more consistent.

### Results

Preprocessing was successfully applied to all files in the dataset. Each file's content was cleaned and stored in a new directory, ensuring that the original dataset remained unaltered. The system randomly selected five preprocessed files to display their content as a verification step, confirming the preprocessing effectiveness.

## 0.2 Creating an Inverted Index

### Approach and Methodology

An inverted index was built to facilitate efficient document retrieval based on individual terms. The index maps each unique word to a set of document names containing that word. The process involves reading preprocessed files, extracting words, and updating the index accordingly.

### Assumptions

- Each word's occurrence is significant for document retrieval, disregarding its specific position within the document.

- The dataset has been preprocessed, meaning it's in a consistent format without punctuation or stopwords, and all text is lowercase.

## Results

The inverted index was successfully created and serialized using Python's `pickle` module for persistent storage. The system supports basic boolean query operations ("AND", "OR", "AND NOT", "OR NOT"), allowing for flexible document retrieval based on user input. The query system demonstrated the ability to accurately return documents that match the specified boolean criteria.

## 0.3 Implementing a Positional Index for Phrase Queries

### Approach and Methodology

The positional index extends the inverted index by additionally storing the position(s) of each word within documents. This structure enables the processing of phrase queries, where the order of words matters. The system preprocesses phrase queries, searches for documents where the phrase occurs in the specified order, and reports the findings.

### Assumptions

- The relevance of a document to a phrase query depends on the occurrence of all query terms in the given order within the document.
- Preprocessed documents are used, ensuring a uniform analysis base.
- Phrase queries are limited to a reasonable length (5 words) for performance considerations.

## Results

The positional index system successfully processes phrase queries, identifying documents where the specified phrases occur exactly. The system's effectiveness was validated by running user-inputted queries and accurately reporting the number of matching documents and their names.

## Conclusion

The project successfully implemented a comprehensive text processing and query system across three major tasks. Each component demonstrated the application of natural language processing techniques and data structures for efficient information retrieval. Through careful preprocessing, strategic index building, and the implementation of robust query functionalities, the system provides a solid foundation for more advanced text analysis and search applications in larger datasets.