Project Report - Text Similarity Checker

1. Introduction

This project implements a simple Text Similarity Checker using Java. It calculates the cosine similarity between two input text documents, identifies common words, and generates a human-readable report highlighting these commonalities.

2. Architecture and Components

The project is structured into several key Java classes, each responsible for a specific part of the text similarity calculation and reporting process:

- **Main.java**: The entry point of the application. It orchestrates the entire process, from reading input files to generating the final report.
- **TextProcessor.java**: Handles the preprocessing of text documents. This includes tokenization, converting text to lowercase, removing punctuation, and filtering out common English stopwords. It also calculates the term frequency for a given list of tokens.
- **CosineSimilarity.java**: Implements the core logic for calculating the cosine similarity between two documents based on their term frequency maps.
- **ReportGenerator.java**: Responsible for formatting and writing the similarity report to a file. It highlights common words in the input documents to visually represent the areas of similarity.

3. Key Functionalities

The Text Similarity Checker performs the following steps:

1 **Text Preprocessing:** Reads two input text files, converts text to lowercase, removes punctuation, splits into tokens, and removes common English stopwords.

2 **Term Frequency Calculation:** Creates a map where keys are unique words and values are their frequencies in each document.

3 **Cosine Similarity Calculation:** Calculates similarity between two term frequency maps using the cosine similarity formula (score 1 = identical, 0 = no similarity).

4 **Report Generation:** Produces a report including similarity percentage and highlights common words between the documents.

4. Usage

To run this project:

1 Ensure you have Java Development Kit (JDK) installed.

2 Place your input text files (e.g., doc1.txt, doc2.txt) in the input/ directory.

Compilation Command:

```
javac -d output src/com/plagiarismchecker/*.java
```

Run Command:

```
java -cp output com.plagiarismchecker.Main
```

The report will be generated in output/report.txt.

5. Example Output

Below is an example of the content found in output/report.txt after running the application with sample inputs:

```
Similarity percentage: 75.00%

--- Document 1 ---
This **project** **belongs** **to** Chandra.

--- Document 2 ---
This **project** **belongs** **to** Charan.
```

6. Conclusion

The Text Similarity Checker provides an effective way to compare two text documents and quantify their similarity using the cosine similarity algorithm. The generated report offers a clear overview of the similarity score and visually highlights shared content, making it easy to understand the commonalities between the documents.