

Sentiment Analysis with Virtual Threads

Student: G00411344

Overview:

This Java application performs sentiment analysis on text files using a lexicon. The application allows users to specify a text file for analysis, configure lexicons, and generate a sentiment report. The application utilises an interface and virtual threads for concurrent processing, enhancing performance during sentiment analysis.

To Run The Application:

The application's main class is Runner. Activate it in the console to run the sentiment analysis application.

The application presents a menu with options for the user:

Specify Files; choose a text file for analysis and optionally specify an output file.

Configure Lexicons; configure the lexicon by entering 'bingliu.txt'.

Execute Analysis; this will generate a sentiment report. The application processes each word in the text file and calculates sentiment values.

Sentiment Report; the application writes the sentiment value to the specified output file (default: ./out.txt).

Exit: Quit the application and a message will inform the user that the application has ended.

How The Application Works:

Runner (Main): Activates the sentiment analysis application by initialising and displaying the menu.

Menu: Manages user interactions, allowing file specification, lexicon configuration, analysis execution, and program termination.

FileProcessor Interface: Defines methods for file processing, including parsing a file and processing text.

LexiconParser: Parses lexicon files and populates a ConcurrentSkipListMap with words and their sentiment values. It supports thread-safe operations.

SentimentParser (Implements FileProcessor): Implements sentiment analysis using a thread pool for parallel processing. Calculates sentiment values based on the lexicon.