Assignment 1: Text Statistics

Subject: Advanced Java

Semester: Fall 2019

Student: Niko Tili

Instructor: Elton Ballhysa­­

Contents

[Introduction 3](#_Toc27031269)

[Solution 3](#_Toc27031270)

[Assumptions 3](#_Toc27031271)

[How to execute 3](#_Toc27031272)

[References 3](#_Toc27031273)

[Appendix 4](#_Toc27031274)

# Introduction

The program is used for computing text statistics. It processes some text files by calculating and displaying following statistics in runtime: number of processed files, number of processing files, five most frequent letters, five most frequent letter pairs, five most frequent words, unigram entropy, bigram entropy, total words and standard deviation of words. The whole processing is done concurrently.

# Solution

Main class al.unyt.edu.advjava.fall2019.assign01.TextStatistics

java al.unyt.edu.advjava.fall2019.assign01.TextStatistics <path> to start the program. The user should specify the folder path. If the path is not valid (does not exist or is not a folder) the program exits. After verifying the path is valid, a Stream containing paths of all .txt files on that folder is created and for each path a single FileConsumer instance, consumes each path as following:

* A thread from a fixed thread pool is started and executes loadFile(Path path)
* In the loadFile(Path path) method, the text file is split into words
* Special characters are removed from each word
* Words are mapped to Word Objects
* For each word unigrams and bigrams are extracted and saved
* If word is not a stop word it is saved too
* Number of words for that file is saved
* Number of processed files is updated

All extracted data is saved in a shared repository, which contains:

* unigramMap (contains each unigram and its frequency)
* bigramMap (contains each bigram and its frequency)
* wordMap (contains each word and its frequency)
* fileWordCountMap (contains each filename and the specific number of words on that file)

all .txt files on that folder are going to be processed concurrently by 50 threads. If the folder contains more than 50 such files, their processing is going to be queued until a thread becomes available. Each thread processes one file at a time. Firstly, the text is split on the whitespaces, all special characters are removed and the words are converted to lowercase. Unigrams and bigrams are extracted from each word. If word is a stop word, it is not included. The statistics are updated. All data from each thread is saved into a shared repository. Distinct unigram, bigram, word and the frequency. Total word count for each file. Total file count.

Every 500 milliseconds statistics are displayed (this is done in another thread which is responsible for terminating the program after files are finished processing).

# Assumptions

The folder does not contain more than 1000 .txt files.  
No order of preference among files.

# How to execute

**java al.unyt.edu.advjava.fall2019.assign01.TextStatistics** **<path>** to start the program. The user should specify the folder path as the first argument. If zero arguments are provided the program exits. If more than one argument is provided, only the first one is considered.

# References

wfgdasfgd

# Appendix