



Consider a text file A, and a pool of text files that contains a number of files. We need a performance-efficient Spring Boot app that reads the path of file A, and the path of the directory that contains the other files from its properties file. Then it needs to read all the files and to assign each file in that pool a score that indicates how much it matches the file A to be able to identify which file of them matches it the most. The score should be based on the similarity between the **words** of both files being compared.

Constraints and Assumptions:

- A chunk of text is only considered a word if it only contains alphabetic characters.
- The ordering of the words is disregarded when comparing two files together, for example, a file consisting of the text “The quick brown fox jumps over the lazy dog” and a file consisting of “The lazy dog jumps over the quick brown fox” are a perfect match.
- If a file in the pool contains exactly all and only the same words in file A, its expected score is 100%.
- If a file in the pool doesn’t contain a single word of all the words existent in file A, its expected score is 0%.
- A file that contains words that don’t exist in file A or doesn’t contain a word or more that are existent in file A isn’t expected to have a 100% score.
- The pool of files can contain from 1 to 20 files.
- Any of the files can contain from 1 to 10M words.