String

Topic: Finding Correlative Words in Text Files

What is a correlated word?

. Refers to pairs of words that are related to each other.

. A typical way to find a correlation is to find a pair of words that are frequently used together in a sentence.

Algorithm to find correlations.

. After storing the frequency for all pairs of words appearing in a sentence, find the k word pairs that appear the most.

. “Sentence means a string separated by ‘.’ ‘?’ ‘!’ However, stop words with less importance such as articles, pronouns, and conjunctions are not included.

. Avoid double counting for the same word pair. For this purpose, all words are changed to lowercase

Input:

File names, number of word pairs? data.txt k

. Composition of input file: English text file

. Number of word pairs: k ← int

Output:

. The k most common characters and their frequencies in the input file excluding stop word.

. k character pairs and frequency counts of most occurrences in the input file, excluding stop word.

. For letter pairs, the smallest word comes first.

. If the frequency is the same, output the words in ascending order (same for word pairs)

2: Submission: HW3.java Only one file is submitted

1: public class HW3 (the rest of the classes in the file are not public)

2: Use default package

3: Delete all comments in the program

4: Korean encoding of Eclipse Workspace is set to MS949

3. Example execution

Example 1 execution result:

```
Filename, number of word pairs? blockchain.txt 3
Tok-k string: blockchains(7) block(6) data(5)
Top-k word pairs: [block, data](3) [bitcoin, blockchain](2) [bitcoin, public](2)
```

Example 2 execution result:

```
Filename, number of word pairs? novel.txt 5
Tok-k string: whale (481) like (289) man (262) ship (239) captain (216)
Top-k word pairs: [sperm, whale] (68) [whale, white] (58) [ahab, captain] (47)
[man, old] (41) [great, whale] (34)
```

Note :

. Will compile with JDK 8

. If there is a serious problem in program configuration or performance, points are deducted regardless of the execution result (indentation, variable or function name, reckless use of Collection objects, etc.)

. Conversely, if the program composition or performance is excellent, it is possible to raise the score.