Distance 0

Procedure followed:-

I have written code to go through each file.

For each file-

1. the program goes through each line.

2. Calculate hashcode for each line.

3. Create hashmap with hashcode as the key and Arraylist with all the lines which has same hashcode.

4. While inserting a new line which has hashcode already existed in the hashmap, check if the line already there in the arraylist. If its there add the line to a set.

5. A set is a datastructure which only keeps unique lines.

6. If the hashcode is not there create a new key in hasmap with hashcode of the line as the key and the ArrayList with this line as the value..

7. Finally the set datastructure contains the duplicate values and write that into a file which is in output distance 0 folder.

Distance 1

Assumptions:

I have considered sentences with distance1 are those where one sentence can be obtained from other sentence by adding or deleting a word from other sentence.

Examples of Distance 1 sentences according to assumptions made are as below:

Lets consider the following sentences which are there in the file

1. I am jagadish
2. I am jagadish bapanapally
3. I bapanapally
4. I jagadish

The output of my code for above lines for distance 1 lines will be as below

1. I am jagadish
2. I am jagadish bapanapally
3. I jagadish.

2nd sentence can be obtained by adding bapanapally to 1st sentence and 4th sentence can be obtained by deleting am from 1st sentence.

Procedure followed to get Distance1 lines:

1. I have written code to go through each file one by one.
2. The program goes through each line of the file.
3. Calculate hashcode for each line.
4. Create hashmap with hashcode as the key and Arraylist with all the lines which has same hashcode.
5. While inserting a new line which has hashcode already existed in the hashmap, check for the string if its already there in the arraylist. If its not there i.e if it's a new value, add the sentence to the arraylist and also to the set.
6. If the hashcode is not there create a new value with hashcode as key in hashmap and the line to the ArrayList and also to the set
7. Finally Set will have all unique sentences.
8. Loop through each line of the set.
9. Delete one word at a time from each line and get the hashcode of that line.
10. Check if there is a sentence with that hashcode in hashmap and if it exists check the line created from 9th step matches with any sentence in the arraylist in hashmap with that hashcode.
11. If it matches add both lines to a new set.
12. Finally write all sentences that are there in the set created at 11th step into a new file.

Execution of the code:

Make sure the folder in which the file “hw1final.java” or “hw1dist1final.java” is present, it contains a folder named “Datafiles” which contains subfolders “25M”,”5M” and “smaller” folders which has the text files.

Distance 0:

Distance 0 code is present in “Distance0code” folder. Before running distance 0 code create a folder “output” in which “hw1final.java” is present.

We can execute the distance 0 code by typing

“javac hw1final.java” and then “java –Xms10M -10240M hw1final” in command prompt.

Distance 1:

Distance 1 code is present in “Distance1code” folder. Before running distance 1 code create a folder “d1output” in which “hw1dist1final.java” is present.

We can execute the distance 1 code by typing

“javac hw1dist1final.java” and then “java –Xms10M -10240M hw1dist1final” in command prompt.

Timings of Distance 0 and distance 1 and number of output lines:

Input file	Distance 0		Distance 1	
	Wallclock time (In milli seconds)	Output lines	Wallclock time (In milli seconds)	Output lines
100.txt	1	2	5	0
1K.txt	3	34	29	0
10K.txt	8	488	216	33
100K.txt	93	7124	2088	978
1M.txt	1180	79902	22892	20311
5M.txt	11094	516778	97431	128225
25M.txt	62024	4432935	350002	499306

Distance 0 Timings also include time for writing output into a file.
Distance 1 Timings doesn’t include time for writing output into a file.