Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 6 REPORT

FURKAN ÖZEV
161044036

Course Assistant: AYSE SERBETCİ TURAN

1 INTRODUCTION

1.1 Problem Definition

In this assignment, we are asked to perform basic Natural Language Processing operations requested with the given dataset. We are asked to design and implement a data structure to hold this data before applying these operations. In order for us to implement data structure, we first need to determine what the data is. This given dataset contains many text files. We are asked to create data structures that holds each word that in all files with their file name and position, in the file. This data structures was requested to be hashmap. To summurize, we will read a text dataset consisting of multiple text files and keep the words that in these files, in the Word HashMap. The key for the word hashmap will be the words and the value will refer to another hashmap that are file hashmap which keeps the occurrences of the word in different files. File hashmap keeps the name of the files which is located, as the key, and the positions of the this file as the value. In addition, each word in the WordMap is required to be connected to each other, so we will increase efficiency in some methods. Also, we can traverse in WordMap efficiently. After obtaining this structure, we will implement two basic operations : retrieving bi-grams and calculating TFIDF values. Bi-gram is simply a piece of text consisting of two sequential words which occurs in a given text at least once. Bi-grams are very informative tools to reveal the semantic relations between words. We are asked to find all the bigram formed by the word given as the starting word. In order to accomplish this, we need to find the word that are after the starting word. We need to search in all the files containing starting word and the occurunce positions in those files and combine these 2 words, after that, keep them in a list data structure. TFIDF is a score which reflects the importance of a word for a single document. A word is informative for a file to be categorized if it occurs frequently in that file but has very few occurrence in other documents in the dataset. In order to calculate the TFIDF value, certain counts should be made and calculations should be made after these counts. To find out how many times t appears in a document, we need to look at the number of occurrences in this file via fileMap. To find out the total number of terms in the document, we need to count how many terms is exist on this file. By proportion these two numbers, we find the term frequency (TF value). To find out the total number of documents, we need to count how many file is exist on dataset. To find out the number of documents with term t in it, we need to look how many file is exist on fileMap of this word. By logarithmic proportion these two numbers, we find the inverse document frequency (IDF value). These two values multiply each other, we

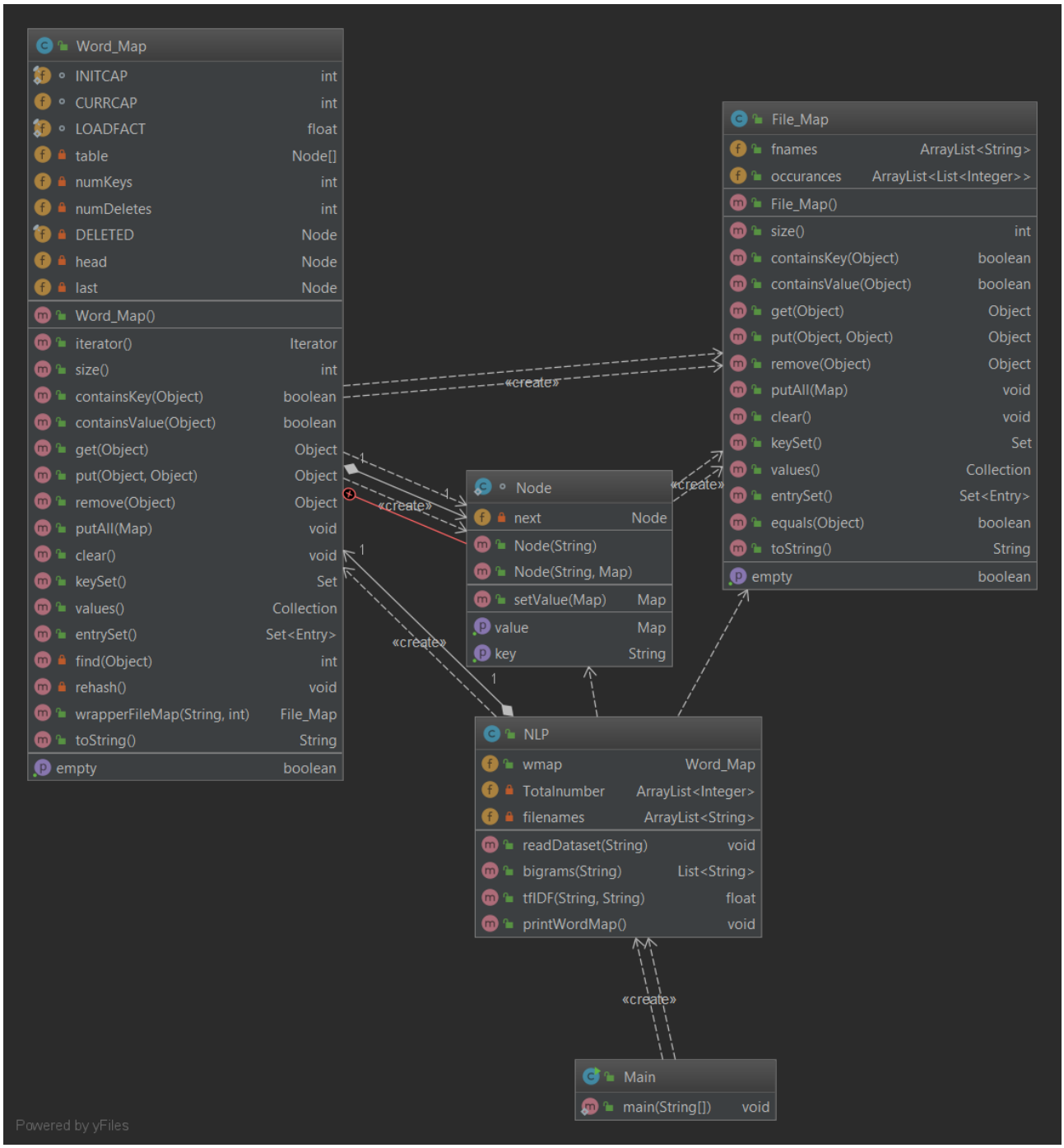
find the Term frequency-inverse document frequency (TFIDF value). Thus, I explained how the requested 2 operations should be performed. We will read information from a file, such as what actions should be performed for what word, and perform it in the program. After each proccess, We will write the result of the proccess on the screen in the desired format. In this way, we will perform the requested some operations in the requested data structure with given dataset. Thus, I have determined all the problems we need to solve in this assignment.

1.2 System Requirements

You must have java installed on your operating system to run programs. Running programs with the current version of Java will be useful for efficiency. Programs can work in both windows, linux and other operating systems installed java. The program was written using jdk-11.0.2 in IntelliJ IDEA. It requires 28,4 kb memory to keep source files. Project size is 1.97 mb with javadoc, dataset folder and input file. In order for the program to work, there should be a folder named dataset in the project folder and text files should be included in dataset folder. In order to run the program, additional input file is required. An input file named input.txt created in the specified format must be located in the project folder to perform the requested operations. For the programs to work properly, dataset folder and input file must be appropriate for the program.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams

An input file named input.txt is required to perform the requested operations. In addition, a data set must be provided for realization these operations. Then the user only runs the program. The results will be seen on the screen. So, Use case diagram is not necessary.

2.3 Problem Solution Approach

I've created 2 HashMap data structures to keep the data as desired. One of these is WordMap. key of WordMap is word (so string) and value of WordMap is another HashMap. The other one is FileMap, which will also be the value of WordMap. Key of FileMap is the file name (so string), value of FileMap is ArrayList holds the integer value. I created a class called Node to keep the key and value in WordMap, and I created Node objects for each word and information. I kept these objects in a Node type sequence. I have created the desired linkedlist structure by linking the Node object I created for each new word to the previous Node object. Compared to the Array, this traversing process is done in a low time complexity. Because the array may be deleted indexes or empty indexes. Let us examine the time complexities of the methods in WordMap. The iterator() method that returns the iterator object of WordMap and the hasNext() and next() methods of the object it returns has the time complexity of O (1). To keep the word and its value, the getKey() , getValue() and setValue() methods of the Node class we created have the O (1) complexity. There is counter that include the number of words put mapping; When you call the size() method, it returns the this counter. Thus, the time complexity is O(1). The time complexity of the isEmpty() method is O (1). Because, it only check same counter whether if zero. n will be number of mapping amount. The time complexity of the find() method is O (n) for worst-case, O(1) for best-case. Because First, the hashcode that are possible index is calculated. It then proceeds from that index until it finds the key or until it reaches the idle index. The time complexity of the containsKey() method is O (n) for worst-case, O(1) for best-case. Because it call find() method then just check it's result. The time complexity of the containsValue() method is O (n). Because, using the linkedlist structure, each value is compared one by one. The time complexity of the get() method is O (n) for worst-case, O(1) for best-case. Because, it call find() method then return value, if it exists. The time complexity of the rehash() method is O (n). Because, it doubles the size of the array and calls the put method again for each element. Since there is no deleted place in the array, we must consider the best case for put. The time complexity of the put() method is O(n) worst-case, O(1) best-case. Because, when the loadfactor value is reached, the rehash() method works for the worst-case. The time complexity of the remove() method is O(n) worst-case O(1) best-case. Because, this method call find() method. It should also remove the object from linked list structure. So if object is exist to remove, it's time complexity is exactly O(n), but object is not exits to remove, the situation is entirely dependent on the find method.

m will number of mapping amount of FileMap to put. The time complexity of the putAll() method is O(nm) worst-case, O(m) best-case. Because, the loop will definitely return m times, we can call it O (m), but it's important for its time complexity that we call the put() in WordMap. put() method can be O (1) or O (n), so putAll() method will be O(m) or O(nm). The time complexity of the clear() method is O(n). Because each index will assigned null. The time complexity of the keySet(), values(), entrySet() methods is O(n). Because it must traverse each mapping via linkedlist structure, Then add object to new structure after, return this structure. Let us examine the FileMap structure. FileMap has 2 ArrayList structures to hold keys and values. ArrayList that is holds the keys holds the strings, and, ArrayList that is holds the value holds the integer lists. Let us examine the time complexities of the methods in FileMap. The time complexity of the size() method is O(1). Because it call ArrayList size() method and it is O(1). The time complexity of the isEmpty() method is O(1). Because it call ArrayList isEmpty() method and it is O(1). The time complexity of the containsKey() and containsValue() methods is O(n). Because, they call ArrayList contains() methods and these contains() methods' time complexity is O(n). Because, The ArrayList structure contains() method, the element makes a comparison with each element until it finds. The time complexity of the get() method is O(n). Because it exactly call containsKey that is O(n) then, if it contain, it call indexOf method of ArrayList. Time complexity of indexOf is O(n). Then return object. So, O(n) + O(n) = O(n). The time complexity of the put() method is O(n) best-case, O(mn) worst-case. Because it exactly call containsKey that is O(n) then, if it contain filename, it adds other occurrences positions(m integer). The time complexity of the remove() method is O(n). Because it exactly call indexOf() method that is O(n) then, if it contain filename, it call remove() and indexOf() methods of ArrayList. remove() and indexOf() method's time complexity is O(n). So, O(n) + O(n) + O(n) = O(n). The time complexity of the putAll() method is O(mn). Because it exactly call put() method for each object(m amount of object). The time complexity of the clear() method is O(n). Because it exactly call clear() method of ArrayList. clear() method's time complexity is O(n). The time complexity of the keySet(), entrySet() methods is O(n). Because it must traverse each mapping via iterator. Then add object to new structure after, return this structure. The time complexity of the values() methods is O(1). Because it only return occurrences ArrayList. The time complexity of the equals() methods is O(1). Because it only check that two object is equal. The time complexity of the toString() methods is O(n). Because it traverse each element. Let us examine the NLP class and its methods. In the readDataSet () method, I opened the folder, opened each file in the folder in turn, and read the words one by one. This is followed by a counter with the position. When he reads each word, he creates

FileMap and puts it in WordMap with the word and FileMap. Since it will be used in tfIDF method, I keep the name of each file I read and the number of terms in that file in ArrayLists. k, get the whole word count. Because the readDataset () method reads all the words one by one, the time complexity of the readDataset () method will be O (k). Let us examine bigrams() method. First of all, it will get FileMap by using get method of WordMap. It then looks at the existence of other words for each file he has found. If other words are in the same file, it checks to see if it comes after it. If it comes, it will create the bigram and add it to the list. n, the number of mapping in WordMap. m, the number of mapping in the file's FileMap. Time complexity of bigram() method is O(nm). Let us examine tfIDF() method. Certain numbers are needed to make the desired calculation. It acquires these numbers by using ArrayList's methods. Then makes the calculation. n, get the mapping number of the word FileMap. Time complexity of tfIDF is O(n). Because, The process with the highest time complexity is it's. I have reached all my goal to solve the problem with HashMap data structure and my own algorithms.

3 RESULT

3.1 Test Cases

I've tested each method one by one after implementing it. I've retested with different scenarios. I tested and checked the words after a few additions to test if the some methods were working correctly. After making sure that they were working properly, I went to test other methods. I have done the same controls from the beginning to the end for the processes that are printed on the screen. I've tried the same actions for different data set and input file when I'm sure there's no problem for this files. When I decided that there was nothing wrong with it, I decided the program was valid. Of course, in all these tests I've encountered certain errors, after correcting these errors I started to test from the beginning. I was convinced that the program was valid.

3.2 Running Results

1.)

```
bigram very
tfidf coffee 0001978
bigram world
bigram costs
bigram is
tfidf Brazil 0000178
```

Result:

```
[very difficult, very soon, very promising, very aggressive, very rapid, very attractive, very vulnerable]

0.0048781727

[world market, world coffee, world made, world share, world price, world markets, world bank, world as, world cocoa, world prices, world for, world tin, world grain]

[costs have, costs and, costs of, costs Transport]

[is the, is not, is possible, is forecast, is expected, is caused, is depending, is at, is estimated, is slightly, is projected, is to, is due, is a, is still, is no, is that, is well, is heading,

0.0073839487


Process finished with exit code 0
```

In Text Editor:

```
[very difficult, very soon, very promising, very aggressive, very rapid, very attractive, very vulnerable]

0.0048781727

[world market, world coffee, world made, world share, world price, world markets, world bank, world as, world cocoa, world prices, world for, world tin, world grain]

[costs have, costs and, costs of, costs Transport]

[is the, is not, is possible, is forecast, is expected, is caused, is depending, is at, is estimated, is slightly, is projected, is to, is due, is a, is still, is no, is that, is
well, is heading, is imperative, is an, is difficult, is time, is too, is keeping, is defining, is sold, is uncertain, is some, is fairly, is unlikely, is willing, is proposing, is
112, is high, is likely, is going, is in, is also, is faced, is basically, is are, is insisting, is unfair, is only, is sending, is planned, is affecting, is trying, is harvested, is
trimming, is improving, is Muda, is meeting, is set, is foreseeable, is beginning, is great, is precisely, is now, is one, is he, is after, is aimed, is committed, is put, is
insufficient, is currently, is wrong, is unrealistic, is it, is often, is being, is searching, is showing, is helping, is why, is apparent, is scheduled, is open, is concerned, is
more, is keen, is how, is downward, is sceptical, is favourable, is unchanged, is passed, is very, is ending, is getting, is down, is flowering]

0.0073839487
```

2.)

```
bigram April
tfidf ICO 0001978
bigram market
bigram exports
bigram limit
tfidf of 0006272
```

Result:

```
[April 1, April one, April if, April he, April they, April registrations, April 2, April and, April 1986, April to, April 6, April 3]

0.011728727

[market prices, market Colombia, market share, market in, market shares, market on, market of, market instead, market is, market should, market and, market last, market forces, market offensive, ma

[exports rising, exports will, exports to, exports as, exports of, exports are, exports last, exports told, exports and, exports which, exports this, exports rose, exports could, exports following,

[limit exports, limit President, limit sales, limit Gilberto, limit responded, limit Since, limit member, limit will, limit for, limit in, limit The, limit can, limit moves, limit Under, limit loss

0.0036331227
```

In Text Editor:

```
[April 1, April one, April if, April he, April they, April registrations, April 2, April and, April 1986, April to, April 6, April 3]

0.011728727

[market prices, market Colombia, market share, market in, market shares, market on, market of, market instead, market is, market should, market and, market last, market forces, market offensive, market he, market Ivorian, market fluctuation, market analysts, market conditions, market would, market this, market at, market Farmers, market circles, market has, market stays, market will, market intervention, market observers, market factors, market trends, market allows, market But, market A, market situation, market earlier, market attracted, market today, market In, market but, market between, market recently, market for, market saw, market watchers, market The, market reality, market structure, market analyses, market oriented, market participation, market Consumers, market that, market could, market factor, market down, market expected, market further, market sometime, market West]

[exports rising, exports will, exports to, exports as, exports of, exports are, exports last, exports told, exports and, exports which, exports this, exports rose, exports could, exports following, exports they, exports fall, exports Loewy, exports about, exports over, exports have, exports were, exports does, exports before, exports or, exports through, exports he, exports in, exports fell, exports The, exports SebaanaKizito, exports served, exports a, exports but, exports by, exports increased, exports low, exports Thailand, exports de]

[limit exports, limit President, limit sales, limit Gilberto, limit responded, limit Since, limit member, limit will, limit for, limit in, limit The, limit can, limit moves, limit Under, limit losses, limit April]

0.0036331227
```

3.)

```
tfidf map 0000764
bigram plans
bigram IBC
tfidf February 0002732
bigram will
bigram balance
tfidf said 0007678
bigram said
```

Result:

```
0.035475325

[plans to, plans yet, plans While, plans and, plans are]

[IBC Jorio, IBC President, IBC will, IBC would, IBC to, IBC said, IBC is, IBC production, IBC warehouses, IBC president, IBC official, IBC left, IBC was, IBC could, IBC purchases, IBC opened, IBC s

0.010707569

[will present, will be, will modify, will affect, will not, will grow, will rise, will also, will increase, will continue, will resume, will meet, will ensue, will have, will reach, will now, will

[balance of, balance earlier]

0.0012537348

[said Distribution, said Resumption, said The, said no, said Tempers, said Their, said Delegates, said If, said Retail, said in, said that, said Coffee, said prospects, said An, said the, said Prod


Process finished with exit code 0
```

In Text Editor:

```
0.035475325

[plans to, plans yet, plans While, plans and, plans are]

[IBC Jorio, IBC President, IBC will, IBC would, IBC to, IBC said, IBC is, IBC production, IBC warehouses, IBC president, IBC official, IBC left, IBC was, IBC could, IBC purchases, IBC opened, IBC statement, IBC tonight, IBC officials, IBC confirmed]

0.010707569

[will present, will be, will modify, will affect, will not, will grow, will rise, will also, will increase, will continue, will resume, will meet, will ensue, will have, will reach, will now, will sink, will begin, will kind, will maintain, will work, will adopt, will give, will go, will benefit, will prevail, will no, will handle, will inevitably, will gain, will suffer, will send, will remain, will almost, will change, will do, will allow, will initially, will heavily, will quickly, will undoubtedly, will react, will ensure, will certainly, will debate, will attend, will then, will call, will resettle, will to, will lose, will agree, will end, will is, will help, will fund, will expand, will host, will permit, will probably, will become, will plummet, will drop, will need, will put, will mount, will decline]

[balance of, balance earlier]

0.0012537348

[said Distribution, said Resumption, said The, said no, said Tempers, said Their, said Delegates, said If, said Retail, said in, said that, said Coffee, said prospects, said An, said the, said Producer, said Opinions, said No, said A, said after, said there, said Producers, said Export, said this, said Prices, said Quotas, said Average, said although, said There, said It, said In, said high, said Administrative, said Earlier, said these, said Arango, said Colombias, said Peru, said Justo, said flood, said Peruvian, said quotas, said they, said it, said stocks, said Some, said objective, said adding, said All, said Other, said a, said one, said Brazil, said Jon, said This, said El, said Within, said West, said He, said assuming, said without, said President, said Questioned, said However, said Teck, said Since, said Major, said Brazils, said consumer, said Dauster, said he, said Commenting, said was, said Colombia, said lower, said But, said commodities, said That, said New, said Roldan, said Tommy, said Were, said on, said Escalante, said At, said yesterday, said today, said when, said Bomani, said producers, said at, said They, said Given, said production, said Exports, said shade, said Indonesia, said Indonesias, said if, said Latin, said world, said stagnating, said IBC, said On, said export, said lows, said Speaking, said Traders, said Customs, said Uganda, said Malaba, said cumulative, said an, said We, said Recent, said ageing, said Growing, said is, said Another, said separately, said Nicaragua, said Zimbabwe, said Madagascar, said those, said Carlos, said have, said his, said efforts, said to, said Buyers, said According, said Marketing, said 126, said Vietnam, said Yesterdays, said except, said according, said India, said One, said Countertrade, said Total, said Early, said Illustrating, said ICO, said Especially, said consuming, said US, said Consumers, said differences, said UK, said Japan, said France, said Canadian, said from, said UIC, said Talks, said several, said Near, said exports, said Agriculture, said Ouko, said High, said more, said With, said Guatemala, said unless, said by, said William, said Debra, said Sandra, said picking, said many, said supplies, said Further, said Gold, said Cattle, said would, said Thailand, said Machakos, said Withdrawals, said Imports, said over, said registrations, said exporter]
```