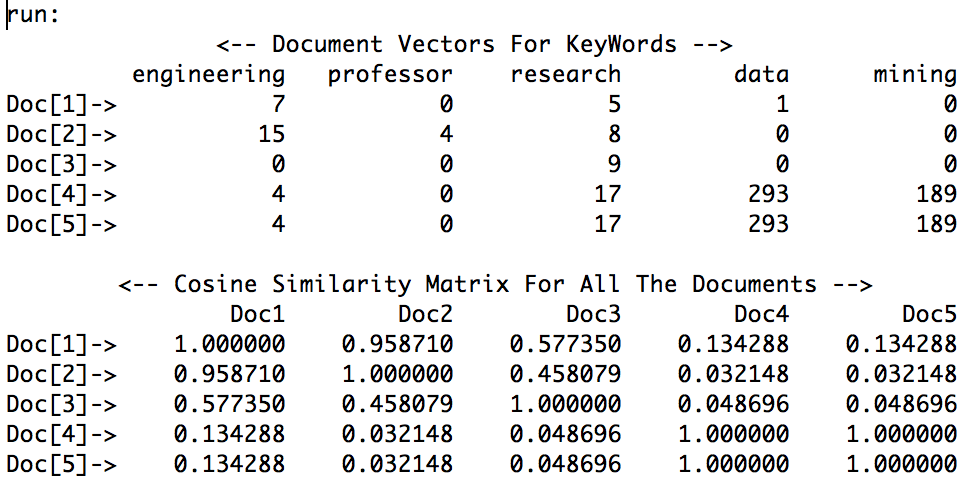
Name: Jiahui Wu

ID: 2670210

Lab #2

Description: Data preprocessing and cosine similarity calculation of webpage txt

Output:



1. Prepare txt file for the word counting.

I opened the listed webpages and copy the text contents to a text editor and save as doc.txt. The five txt files were saved in a folder /Users/jiahuiwu/Desktop/CIS660/lab2/doc.

<http://www.csuohio.edu/engineering/eecs/faculty-staff>

<http://engineering.case.edu/eecs/>

<http://my.clevelandclinic.org/research>

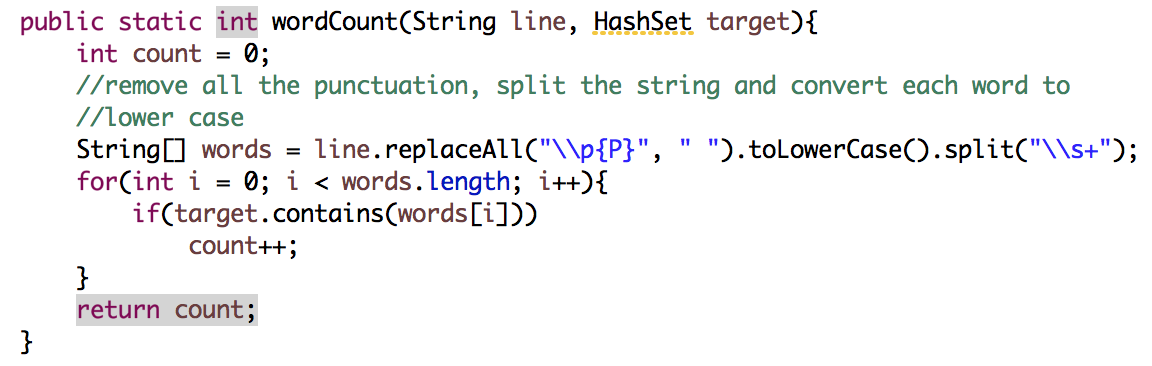
<https://en.wikipedia.org/wiki/Data_mining>

<https://en.wikipedia.org/wiki/Data_mining#Data_mining>

2. Programming to count word. I use JAVA (software Eclipse) to do the coding part.

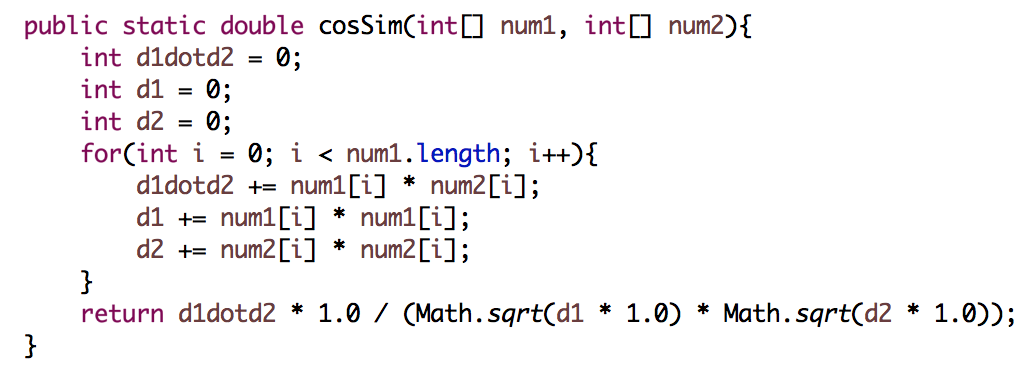
2-1. public int wordCount(String line, HashSet target)

First, I create a method to count the word occurrence, the input arguments are a string and a set, the set contains a set of keywords (the keyword and the other words stemmed from the keyword). First, I replace all the punctuation with white space and convert the string to lower case, then split the string with white space. If the set contains the word the pointer is pointing to, the word count will add up. It returns the word count.



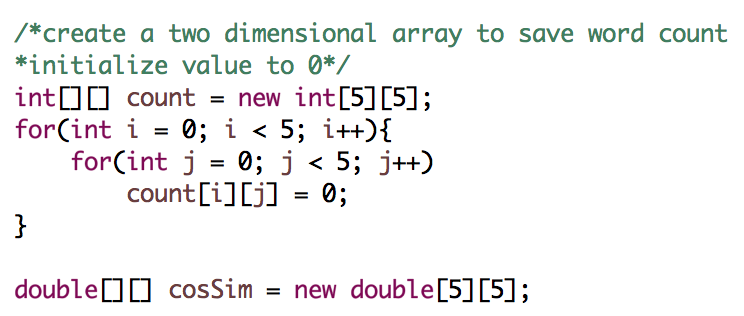
2-2. public double coSim(int[] num1, int[] num2)

Another method is to calculate the cosine similarity. Input arguments are two integer arrays, representing the word count the different key words between two files. It returns the cosine similarity of the two arrays.

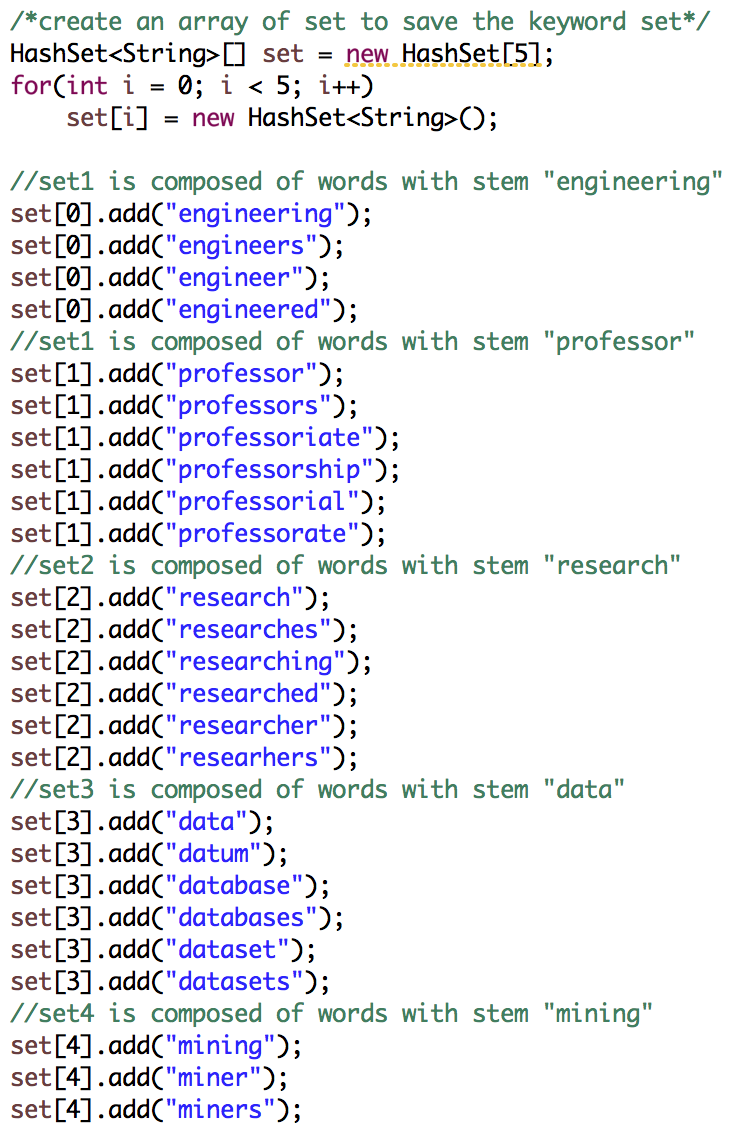


2-3. Main function

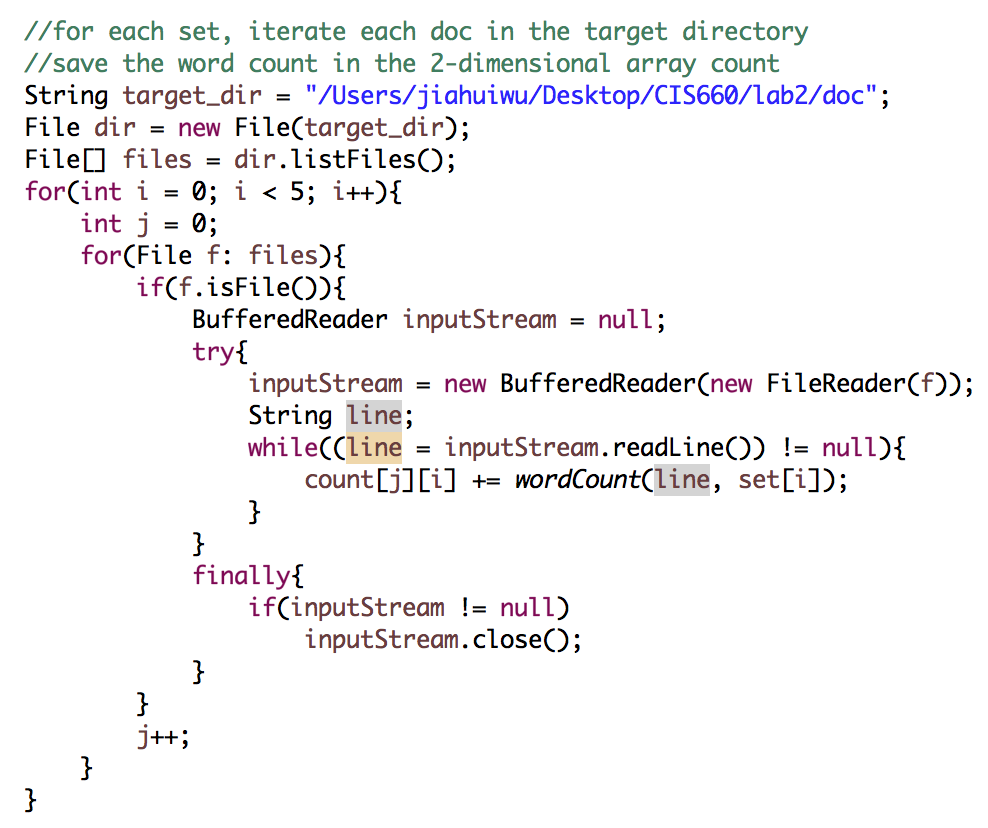
2-3-1. Create two two-dimensional array to save the word count (initialize it to 0) and cosine similarity.



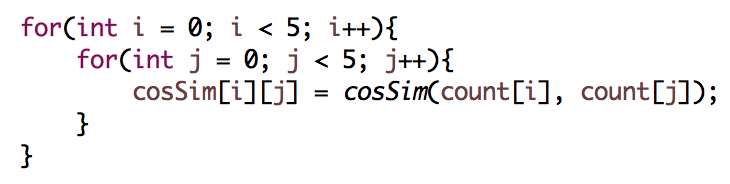
2-3-2. Establish sets of keyword. Create a list of hashset, in each set, add the keyword and related words to the set.



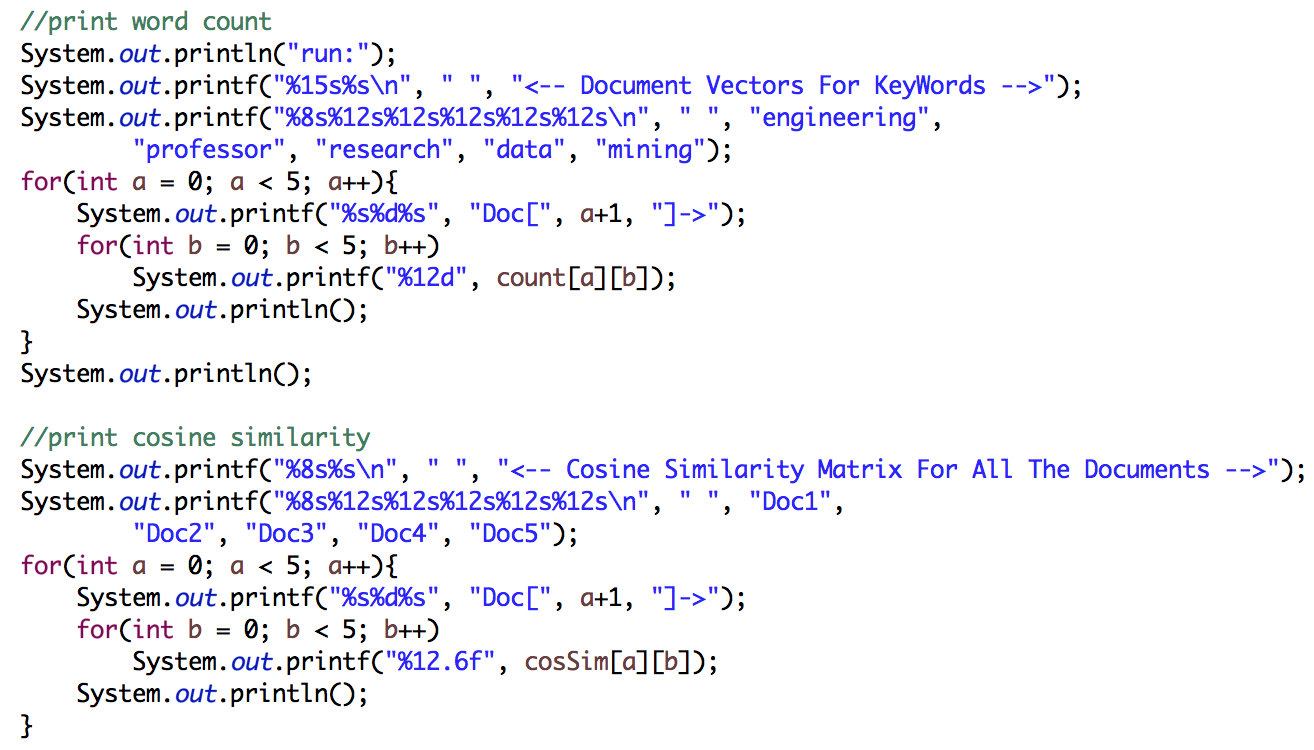
2-3-3. For each set, iterate each doc in the target folder of the directory, for each count[i], it contains the word count of each key word.



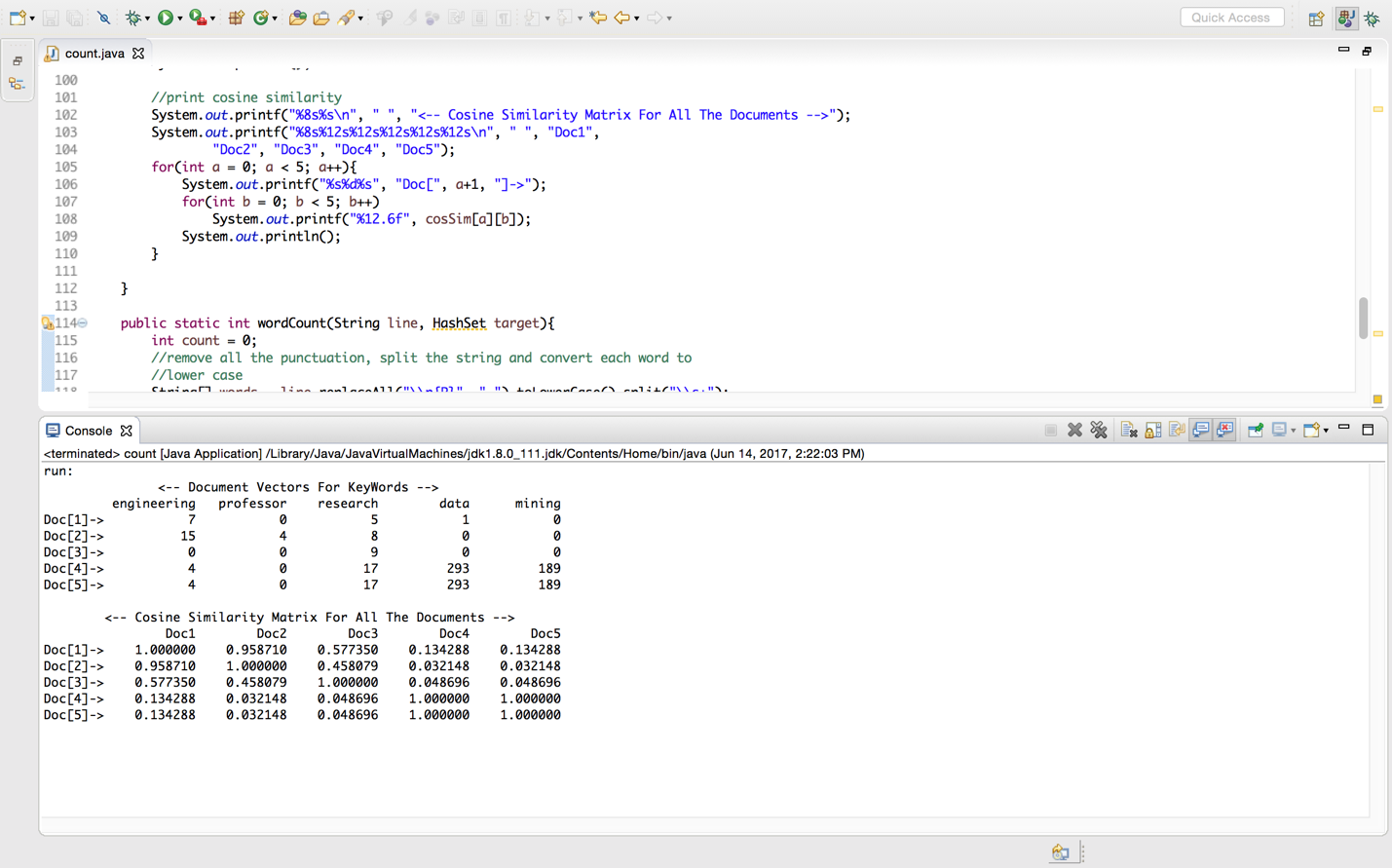
2-3-4. For each pair, calculate the cosine similarity.



2-3-5. Print the word count and cosine similarities.



3. Run and get result (enlarged output is in the first page)



CODE:

import java.io.FileReader;

import java.io.BufferedReader;

import java.io.IOException;

import java.util.HashSet;

import java.io.File;

public class count {

public static void main(String[] args) throws IOException {

/\*create a two dimensional array to save word count

\*initialize value to 0\*/

int[][] count = new int[5][5];

for(int i = 0; i < 5; i++){

for(int j = 0; j < 5; j++)

count[i][j] = 0;

}

double[][] cosSim = new double[5][5];

/\*create an array of set to save the keyword set\*/

HashSet<String>[] set = new HashSet[5];

for(int i = 0; i < 5; i++)

set[i] = new HashSet<String>();

//set1 is composed of words with stem "engineering"

set[0].add("engineering");

set[0].add("engineers");

set[0].add("engineer");

set[0].add("engineered");

//set1 is composed of words with stem "professor"

set[1].add("professor");

set[1].add("professors");

set[1].add("professoriate");

set[1].add("professorship");

set[1].add("professorial");

set[1].add("professorate");

//set2 is composed of words with stem "research"

set[2].add("research");

set[2].add("researches");

set[2].add("researching");

set[2].add("researched");

set[2].add("researcher");

set[2].add("researhers");

//set3 is composed of words with stem "data"

set[3].add("data");

set[3].add("datum");

set[3].add("database");

set[3].add("databases");

set[3].add("dataset");

set[3].add("datasets");

//set4 is composed of words with stem "mining"

set[4].add("mining");

set[4].add("miner");

set[4].add("miners");

//for each set, iterate each doc in the target directory

//save the word count in the 2-dimensional array count

String target\_dir = "/Users/jiahuiwu/Desktop/CIS660/lab2/doc";

File dir = new File(target\_dir);

File[] files = dir.listFiles();

for(int i = 0; i < 5; i++){

int j = 0;

for(File f: files){

if(f.isFile()){

BufferedReader inputStream = null;

try{

inputStream = new BufferedReader(new FileReader(f));

String line;

while((line = inputStream.readLine()) != null){

count[j][i] += wordCount(line, set[i]);

}

}

finally{

if(inputStream != null)

inputStream.close();

}

}

j++;

}

}

for(int i = 0; i < 5; i++){

for(int j = 0; j < 5; j++){

cosSim[i][j] = cosSim(count[i], count[j]);

}

}

//print word count

System.out.println("run:");

System.out.printf("%15s%s\n", " ", "<-- Document Vectors For KeyWords -->");

System.out.printf("%8s%12s%12s%12s%12s%12s\n", " ", "engineering",

"professor", "research", "data", "mining");

for(int a = 0; a < 5; a++){

System.out.printf("%s%d%s", "Doc[", a+1, "]->");

for(int b = 0; b < 5; b++)

System.out.printf("%12d", count[a][b]);

System.out.println();

}

System.out.println();

//print cosine similarity

System.out.printf("%8s%s\n", " ", "<-- Cosine Similarity Matrix For All The Documents -->");

System.out.printf("%8s%12s%12s%12s%12s%12s\n", " ", "Doc1",

"Doc2", "Doc3", "Doc4", "Doc5");

for(int a = 0; a < 5; a++){

System.out.printf("%s%d%s", "Doc[", a+1, "]->");

for(int b = 0; b < 5; b++)

System.out.printf("%12.6f", cosSim[a][b]);

System.out.println();

}

}

public static int wordCount(String line, HashSet target){

int count = 0;

//remove all the punctuation, split the string and convert each word to

//lower case

String[] words = line.replaceAll("\\p{P}", " ").toLowerCase().split("\\s+");

for(int i = 0; i < words.length; i++){

if(target.contains(words[i]))

count++;

}

return count;

}

public static double cosSim(int[] num1, int[] num2){

int d1dotd2 = 0;

int d1 = 0;

int d2 = 0;

for(int i = 0; i < num1.length; i++){

d1dotd2 += num1[i] \* num2[i];

d1 += num1[i] \* num1[i];

d2 += num2[i] \* num2[i];

}

return d1dotd2 \* 1.0 / (Math.sqrt(d1 \* 1.0) \* Math.sqrt(d2 \* 1.0));

}

}