Rakesh Harish **rharish@uncc.edu**
800984018

Assignment 2

This programming assignment submission consists of 4 Java files DocWordCount.java, TermFrequency.java, TFIDF.java and Search.java. with its corresponding output files - DocWordCount.out, TermFrequency.out, TFIDF.out.

For search.java –

a. query1: “computer science”, and name it query1.out

b. query2: “data analysis”, and name it query2.out

NOTE for the users of these programs: Prior to running each program make sure the correct input path is given for the program and the output path folder is not already present in the hdfs. If present please remove it and its contents with same names.

Instructions:

- Place the DocWordCount.java, TermFrequency.java, TFIDF.java, Search.java and Rank.java in one folder.
- create a subfolder with the name input and copy all the input files in that sub folder.
- Open the terminal in the .java files’ folder location and run the below commands to remove the existing folder if present (because of usage of others’ programs with same names) and to copy the provided 8 input files to the HDFS’ input folder.

  hadoop fs -rm -r input

  hadoop fs -put input/* input

- Run the below code to remove any existing folder structure in the HDFS for the outputs (because of usage of others’ programs with same names)

  hadoop fs -rm -r inputtemp

  hadoop fs -rm -r outputDocWordCount

  hadoop fs -rm -r outputTermFrequency

  hadoop fs -rm -r outputTFID

  hadoop fs -rm -r outputSearch

  hadoop fs -rm -r outputRank

Rakesh Harish **rharish@uncc.edu**
800984018

- Run the below commands to remove and create new output folder if present to extract the outputs generated from the HDFS

  rm -r outputs

  mkdir -p outputs

Note: Remember to delete the build and jar files before running these cmds each time for a particular .java file…

DocWordCount.java

rm -r build

rm docwordcount.jar

mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* DocWordCount.java -d build -Xlint

jar -cvf docwordcount.jar -C build/ .

hadoop jar docwordcount.jar org.myorg.DocWordCount input outputDocWordCount

hadoop fs -get outputDocWordCount/part-r-00000 outputs/DocWordCountOutput.out

TermFrequency.java

rm -r build

rm termfrequency.jar

mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* TermFrequency.java -d build -Xlint

jar -cvf termfrequency.jar -C build/ .

hadoop jar termfrequency.jar org.myorg.TermFrequency input outputTermFrequency

hadoop fs -get outputTermFrequency/part-r-00000 outputs/TermFrequencyOutput.out

Rakesh Harish **rharish@uncc.edu**
800984018

TFIDF.java

rm -r build

rm tfidf.jar

mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* TermFrequency.java TFIDF.java -d build -Xlint

jar -cvf tfidf.jar -C build/ .

hadoop jar tfidf.jar org.myorg.TFIDF input outputTFID

hadoop fs -get outputTFID/part-r-00000 outputs/TFIDFOutput.out

Search.java

rm -r build

rm search.jar

mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* Search.java -d build -Xlint

jar -cvf search.jar -C build/ .

hadoop jar search.jar org.myorg.Search outputTFID outputSearch <search query>

- <search query> = computer science or data analysis
- Eg: hadoop jar search.jar org.myorg.Search outputTFID outputSearch data analysis

hadoop fs -get outputSearch/part-r-00000 outputs/query1.out

hadoop fs -get outputSearch/part-r-00000 outputs/query2.out

query1 for computer science

query2 for data analysis