Generic Explanation:

This assignment required multiple things such as file handling, string parsing etc. I am using file handling for *positive.txt and negative.txt.* Whereas the input file (*Comments.txt*) is being read by the mapper function itself. I am converting the whole line into a string and then parsing the string by storing the string into a different variable. For this I hardcoded the parser code as I saw the input file and before every comment, there is always a "T".

Another implementation was that in order to see if the reducer has ended, I am using a built-in function Cleaner (). This function is basically called when the reducer has finished its tasks.

Furthermore, for this assignment I am using HDFS as for some reason when I installed HDFS, my Hadoop-standalone stopped working.

Rest of the implementation is requirement specific explained below:

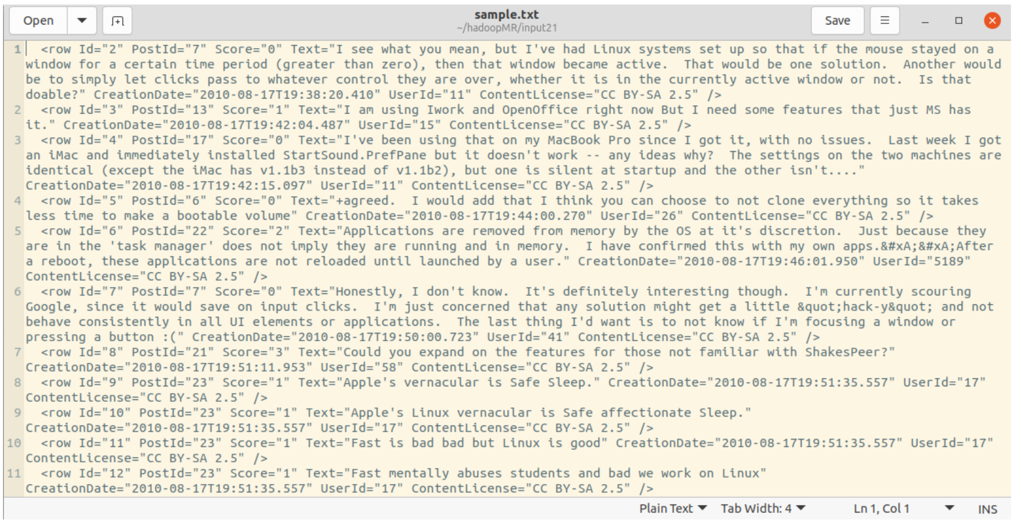
Keyword Sentiment Analysis:

For this task I first read both positive.txt and negative.txt into respective strings and then parsed them into an array of strings such that each index will contain a whole word.

Similarly, I am reading the keyword from "keyword.txt". Then I am searching that keyword in every comment. If a comment has that particular keyword, then all of its contents will be parsed into an array of strings, each index containing a word of that comment. Then each word is compared with both the files and depending upon the number of instances in each file, respective values are updated. In the end, the values are compared and if the comment has more instances in positive, then 1 is added to "overall" (a variable that stores the overall sentiment of the word) and vice versa. In the end, the variable overall is compared and if it is :

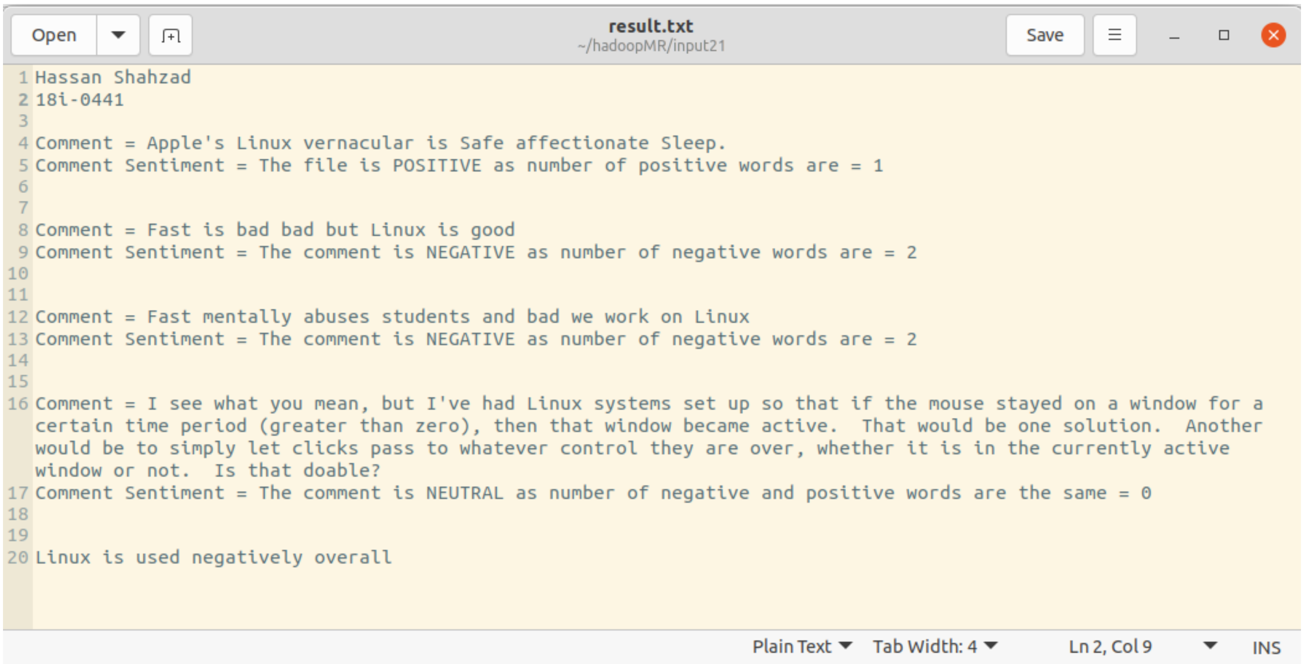
- >0 := This means that the number of positive comments were more and overall sentiment is positive.
- <0 := This means that the number of negative comments were more and overall sentiment is negative.
- =0 := This means that the number of positive and negative comments are equal and overall sentiment is neutral.

Hence, my implementation prints sentiment of each comment along with the overall sentiment. I ran it on a sample dataset to test it out first. For testing, the keyword used was "Linux". The screenshots of sample dataset and its outputs are given as follows:



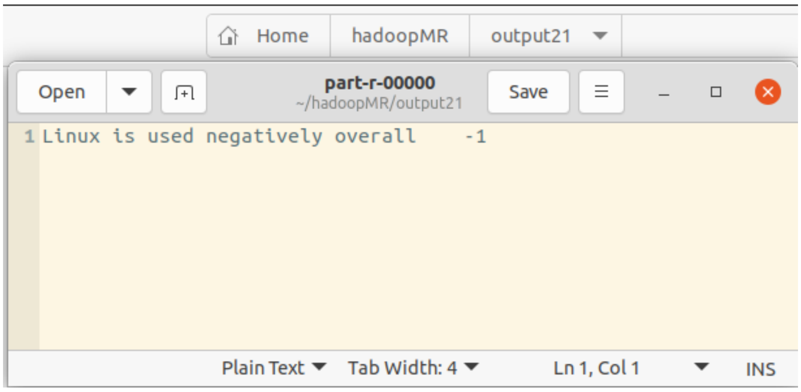
```
 <row Id="2" PostId="7" Score="0" Text="I see what you mean, but I've had Linux systems set up so that if the mouse stayed on a window for a certain time period (greater than zero), then that window became active.  That would be one solution.  Another would be to simply let clicks pass to whatever control they are over, whether it is in the currently active window or not.  Is that doable?" CreationDate="2010-08-17T19:38:20.410" UserId="11" ContentLicense="CC BY-SA 2.5" />
 <row Id="3" PostId="13" Score="1" Text="I am using Iwork and OpenOffice right now But I need some features that just MS has it." CreationDate="2010-08-17T19:42:04.487" UserId="15" ContentLicense="CC BY-SA 2.5" />
 <row Id="4" PostId="17" Score="0" Text="I've been using that on my MacBook Pro since I got it, with no issues.  Last week I got an iMac and immediately installed StartSound.PrefPane but it doesn't work -- any ideas why?  The settings on the two machines are identical (except the iMac has v1.1b3 instead of v1.1b2), but one is silent at startup and the other isn't...." CreationDate="2010-08-17T19:42:15.097" UserId="11" ContentLicense="CC BY-SA 2.5" />
 <row Id="5" PostId="6" Score="0" Text="+agreed.  I would add that I think you can choose to not clone everything so it takes less time to make a bootable volume" CreationDate="2010-08-17T19:44:00.270" UserId="26" ContentLicense="CC BY-SA 2.5" />
 <row Id="6" PostId="22" Score="2" Text="Applications are removed from memory by the OS at it's discretion.  Just because they are in the 'task manager' does not imply they are running and in memory.  I have confirmed this with my own apps.&#xA;&#xA;After a reboot, these applications are not reloaded until launched by a user." CreationDate="2010-08-17T19:46:01.950" UserId="5189" ContentLicense="CC BY-SA 2.5" />
 <row Id="7" PostId="7" Score="0" Text="Honestly, I don't know.  It's definitely interesting though.  I'm currently scouring Google, since it would save on input clicks.  I'm just concerned that any solution might get a little &quot;hack-y&quot; and not behave consistently in all UI elements or applications.  The last thing I'd want is to not know if I'm focusing a window or pressing a button :(" CreationDate="2010-08-17T19:50:00.723" UserId="41" ContentLicense="CC BY-SA 2.5" />
 <row Id="8" PostId="21" Score="3" Text="Could you expand on the features for those not familiar with ShakesPeer?" CreationDate="2010-08-17T19:51:11.953" UserId="58" ContentLicense="CC BY-SA 2.5" />
 <row Id="9" PostId="23" Score="1" Text="Apple's vernacular is Safe Sleep." CreationDate="2010-08-17T19:51:35.557" UserId="17" ContentLicense="CC BY-SA 2.5" />
 <row Id="10" PostId="23" Score="1" Text="Apple's Linux vernacular is Safe affectionate Sleep." CreationDate="2010-08-17T19:51:35.557" UserId="17" ContentLicense="CC BY-SA 2.5" />
 <row Id="11" PostId="23" Score="1" Text="Fast is bad bad but Linux is good" CreationDate="2010-08-17T19:51:35.557" UserId="17" ContentLicense="CC BY-SA 2.5" />
 <row Id="12" PostId="23" Score="1" Text="Fast mentally abuses students and bad we work on Linux" CreationDate="2010-08-17T19:51:35.557" UserId="17" ContentLicense="CC BY-SA 2.5" />
```

Fig 3.1: Test Dataset



```
Hassan Shahzad
18i-0441

Comment = Apple's Linux vernacular is Safe affectionate Sleep.
Comment Sentiment = The file is POSITIVE as number of positive words are = 1


Comment = Fast is bad bad but Linux is good
Comment Sentiment = The comment is NEGATIVE as number of negative words are = 2


Comment = Fast mentally abuses students and bad we work on Linux
Comment Sentiment = The comment is NEGATIVE as number of negative words are = 2


Comment = I see what you mean, but I've had Linux systems set up so that if the mouse stayed on a window for a
certain time period (greater than zero), then that window became active.  That would be one solution.  Another
would be to simply let clicks pass to whatever control they are over, whether it is in the currently active
window or not.  Is that doable?
Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = 0


Linux is used negatively overall
```

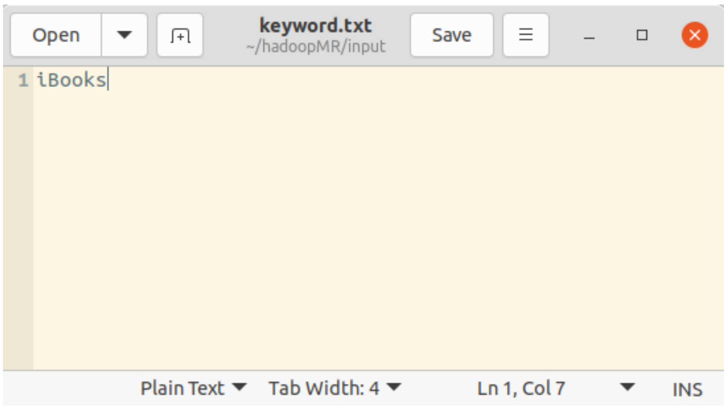
Fig 3.2: Test output (Customized)



```
Linux is used negatively overall    -1
```

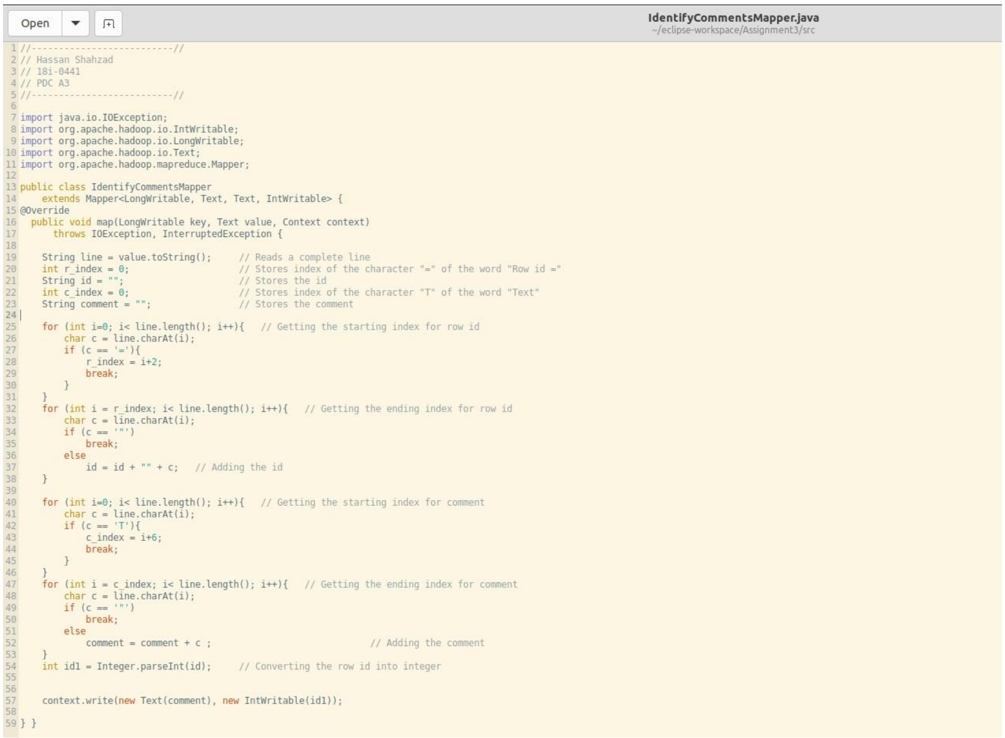
Fig 3.3: Test Output

After this, the same code was run on the original dataset. The code that was run and the screenshots of its outputs are given as follows:



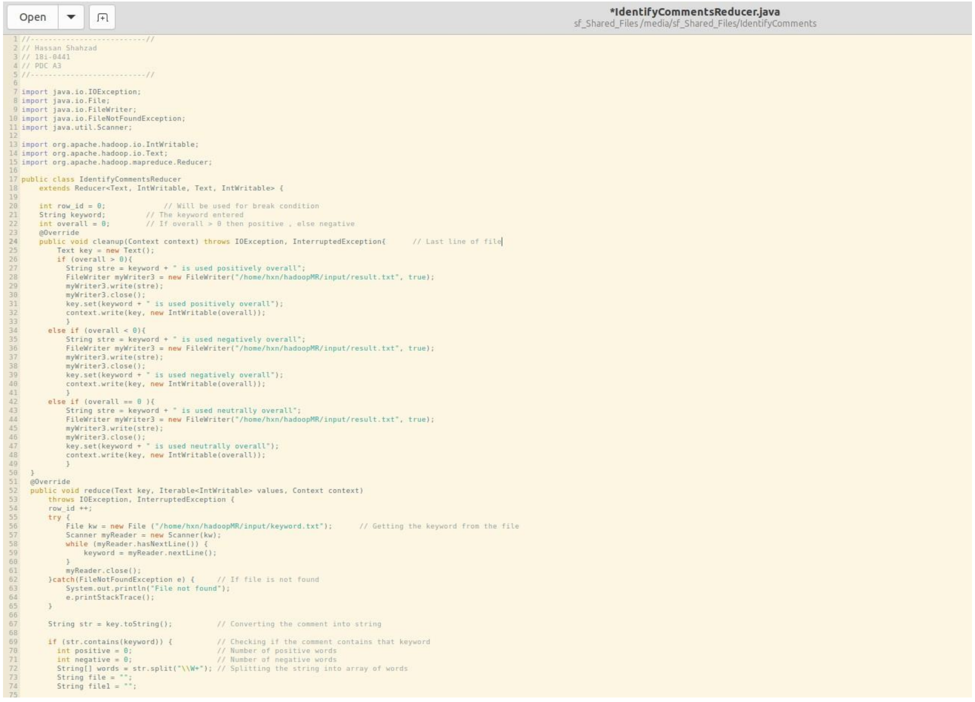
```
iBooks
```

Fig 3.4: Keyword File


Open
IdentifyCommentsMapper.java
~/eclipse-workspace/Assignment3/src

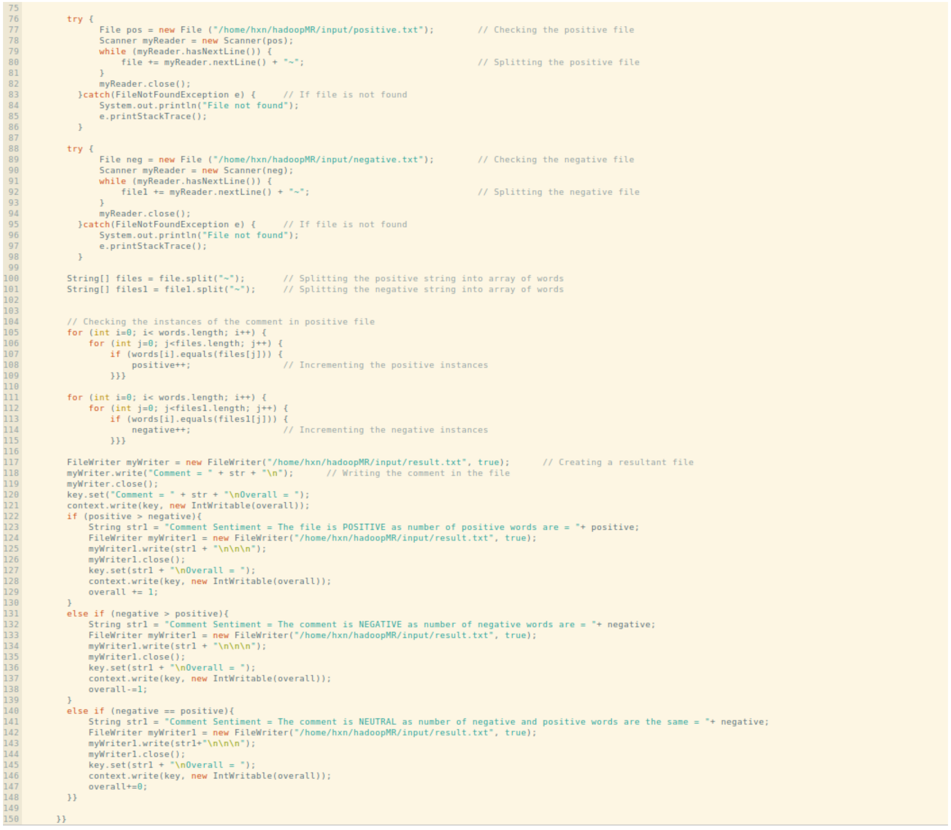
```java
//-------------------------//
// Hassan Shahzad
// 18i-0441
// PDC A3
//-------------------------//

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class IdentifyCommentsMapper
    extends Mapper<LongWritable, Text, Text, IntWritable> {
@Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {

    String line = value.toString();     // Reads a complete line
    int r_index = 0;                    // Stores index of the character "=" of the word "Row id ="
    String id = "";                     // Stores the id
    int c_index = 0;                    // Stores index of the character "T" of the word "Text"
    String comment = "";                // Stores the comment

    for (int i=0; i< line.length(); i++){   // Getting the starting index for row id
        char c = line.charAt(i);
        if (c == '='){
            r_index = i+2;
            break;
        }
    }
    for (int i = r_index; i< line.length(); i++){   // Getting the ending index for row id
        char c = line.charAt(i);
        if (c == '"')
            break;
        else
            id = id + "" + c;   // Adding the id
    }

    for (int i=0; i< line.length(); i++){   // Getting the starting index for comment
        char c = line.charAt(i);
        if (c == 'T'){
            c_index = i+6;
            break;
        }
    }
    for (int i = c_index; i< line.length(); i++){   // Getting the ending index for comment
        char c = line.charAt(i);
        if (c == '"')
            break;
        else
            comment = comment + c ;                             // Adding the comment
    }
    int id1 = Integer.parseInt(id);     // Converting the row id into integer


    context.write(new Text(comment), new IntWritable(id1));

} }
```

Fig 3.5: Mapper Class


Open
*IdentifyCommentsReducer.java
sf_Shared_Files /media/sf_Shared_Files/IdentifyComments

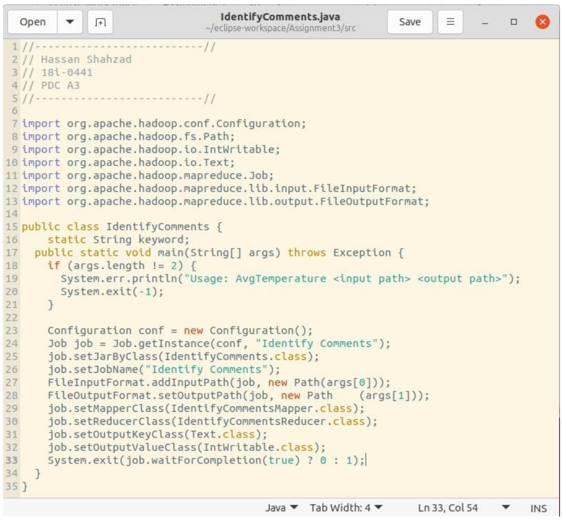
```java
//-------------------------//
// Hassan Shahzad
// 18i-0441
// PDC A3
//-------------------------//

import java.io.IOException;
import java.io.File;
import java.io.FileWriter;
import java.io.FileNotFoundException;
import java.util.Scanner;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class IdentifyCommentsReducer
    extends Reducer<Text, IntWritable, Text, IntWritable> {

    int row_id = 0;              // Will be used for break condition
    String keyword;          // The keyword entered
    int overall = 0;         // If overall > 0 then positive , else negative
    @Override
    public void cleanup(Context context) throws IOException, InterruptedException{      // Last line of file
        Text key = new Text();
        if (overall > 0){
          String stre = keyword + " is used positively overall";
          FileWriter myWriter3 = new FileWriter("/home/hxn/hadoopMR/input/result.txt", true);
          myWriter3.write(stre);
          myWriter3.close();
          key.set(keyword + " is used positively overall");
          context.write(key, new IntWritable(overall));
          }
      else if (overall < 0){
          String stre = keyword + " is used negatively overall";
          FileWriter myWriter3 = new FileWriter("/home/hxn/hadoopMR/input/result.txt", true);
          myWriter3.write(stre);
          myWriter3.close();
          key.set(keyword + " is used negatively overall");
          context.write(key, new IntWritable(overall));
          }
      else if (overall == 0 ){
          String stre = keyword + " is used neutrally overall";
          FileWriter myWriter3 = new FileWriter("/home/hxn/hadoopMR/input/result.txt", true);
          myWriter3.write(stre);
          myWriter3.close();
          key.set(keyword + " is used neutrally overall");
          context.write(key, new IntWritable(overall));
          }
  }
  @Override
  public void reduce(Text key, Iterable<IntWritable> values, Context context)
      throws IOException, InterruptedException {
      row_id ++;
      try {
          File kw = new File ("/home/hxn/hadoopMR/input/keyword.txt");      // Getting the keyword from the file
          Scanner myReader = new Scanner(kw);
          while (myReader.hasNextLine()) {
              keyword = myReader.nextLine();
          }
          myReader.close();
      }catch(FileNotFoundException e) {     // If file is not found
          System.out.println("File not found");
          e.printStackTrace();
      }

      String str = key.toString();          // Converting the comment into string

      if (str.contains(keyword)) {          // Checking if the comment contains that keyword
        int positive = 0;                   // Number of positive words
        int negative = 0;                   // Number of negative words
        String[] words = str.split("\\W+"); // Splitting the string into array of words
        String file = "";
        String file1 = "";

```

Fig 3.6: Reducer Class (Pt-1)



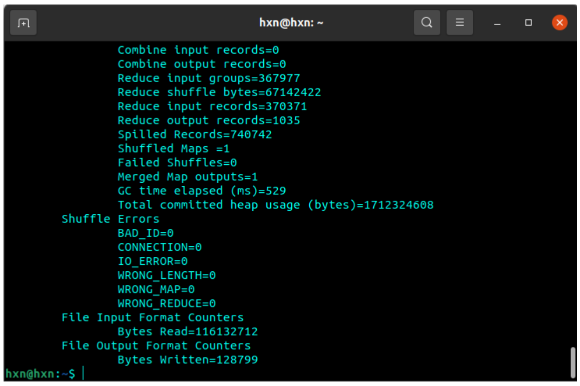
```java

        try {
              File pos = new File ("/home/hxn/hadoopMR/input/positive.txt");        // Checking the positive file
              Scanner myReader = new Scanner(pos);
              while (myReader.hasNextLine()) {
                  file += myReader.nextLine() + "~";                                // Splitting the positive file
              }
              myReader.close();
          }catch(FileNotFoundException e) {     // If file is not found
              System.out.println("File not found");
              e.printStackTrace();
          }

        try {
              File neg = new File ("/home/hxn/hadoopMR/input/negative.txt");        // Checking the negative file
              Scanner myReader = new Scanner(neg);
              while (myReader.hasNextLine()) {
                  file1 += myReader.nextLine() + "~";                               // Splitting the negative file
              }
              myReader.close();
          }catch(FileNotFoundException e) {     // If file is not found
              System.out.println("File not found");
              e.printStackTrace();
          }

        String[] files = file.split("~");       // Splitting the positive string into array of words
        String[] files1 = file1.split("~");     // Splitting the negative string into array of words


        // Checking the instances of the comment in positive file
        for (int i=0; i< words.length; i++) {
            for (int j=0; j<files.length; j++) {
                if (words[i].equals(files[j])) {
                    positive++;                 // Incrementing the positive instances
                }}}

        for (int i=0; i< words.length; i++) {
            for (int j=0; j<files1.length; j++) {
                if (words[i].equals(files1[j])) {
                    negative++;                 // Incrementing the negative instances
                }}}

        FileWriter myWriter = new FileWriter("/home/hxn/hadoopMR/input/result.txt", true);      // Creating a resultant file
        myWriter.write("Comment = " + str + "\n");      // Writing the comment in the file
        myWriter.close();
        key.set("Comment = " + str + "\nOverall = ");
        context.write(key, new IntWritable(overall));
        if (positive > negative){
            String str1 = "Comment Sentiment = The file is POSITIVE as number of positive words are = "+ positive;
            FileWriter myWriter1 = new FileWriter("/home/hxn/hadoopMR/input/result.txt", true);
            myWriter1.write(str1 + "\n\n\n");
            myWriter1.close();
            key.set(str1 + "\nOverall = ");
            context.write(key, new IntWritable(overall));
            overall += 1;
        }
        else if (negative > positive){
            String str1 = "Comment Sentiment = The comment is NEGATIVE as number of negative words are = "+ negative;
            FileWriter myWriter1 = new FileWriter("/home/hxn/hadoopMR/input/result.txt", true);
            myWriter1.write(str1 + "\n\n\n");
            myWriter1.close();
            key.set(str1 + "\nOverall = ");
            context.write(key, new IntWritable(overall));
            overall-=1;
        }
        else if (negative == positive){
            String str1 = "Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = "+ negative;
            FileWriter myWriter1 = new FileWriter("/home/hxn/hadoopMR/input/result.txt", true);
            myWriter1.write(str1+"\n\n\n");
            myWriter1.close();
            key.set(str1 + "\nOverall = ");
            context.write(key, new IntWritable(overall));
            overall+=0;
        }}

      }}
```

Fig 3.7: Reducer Class (Pt-2)



```java
//-------------------------//
// Hassan Shahzad
// 18i-0441
// PDC A3
//-------------------------//

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class IdentifyComments {
    static String keyword;
  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Usage: AvgTemperature <input path> <output path>");
      System.exit(-1);
    }

    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Identify Comments");
    job.setJarByClass(IdentifyComments.class);
    job.setJobName("Identify Comments");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path    (args[1]));
    job.setMapperClass(IdentifyCommentsMapper.class);
    job.setReducerClass(IdentifyCommentsReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Fig 3.8: Main Java File



```
                Combine input records=0
                Combine output records=0
                Reduce input groups=367977
                Reduce shuffle bytes=67142422
                Reduce input records=370371
                Reduce output records=1035
                Spilled Records=740742
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=529
                Total committed heap usage (bytes)=1712324608
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=116132712
        File Output Format Counters
                Bytes Written=128799
hxn@hxn:~$ 
```

Fig 3.9: Execution of the Code

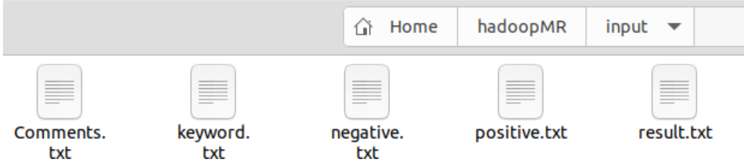
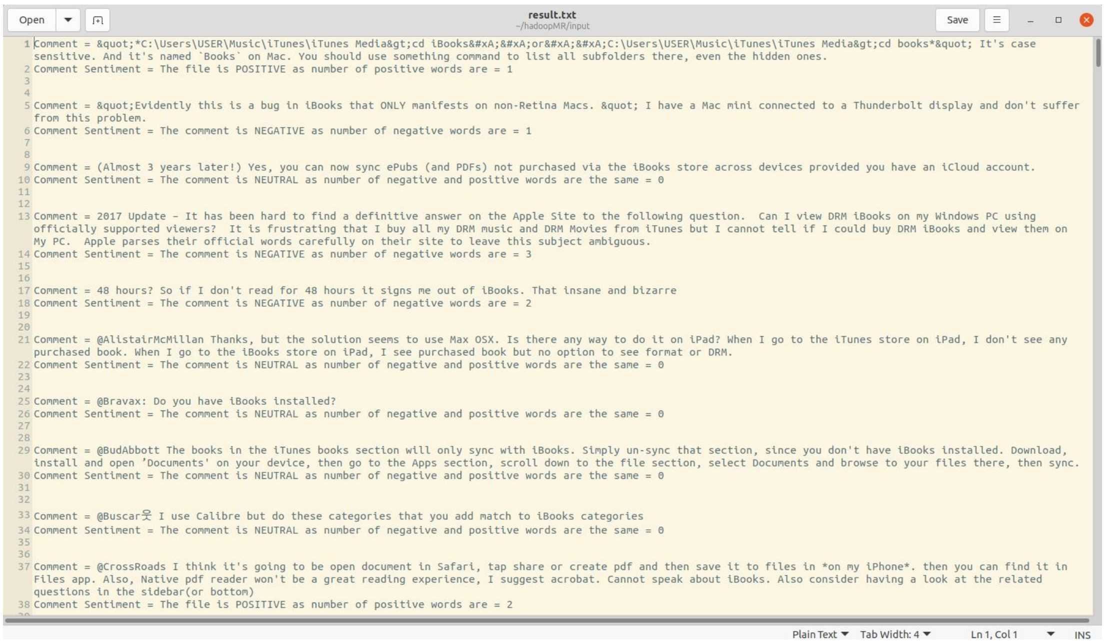
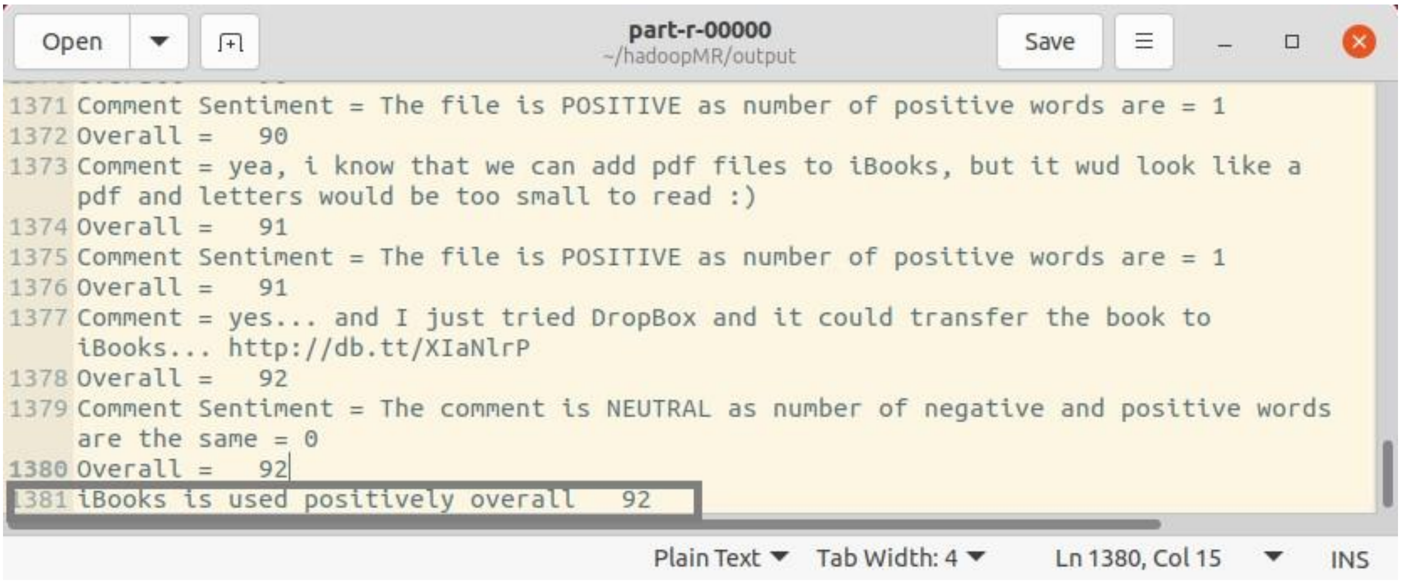


Fig 3.10: Input Folder



```
Comment = &quot;*C:\Users\USER\Music\iTunes\iTunes Media&gt;cd iBooks&#xA;&#xA;or&#xA;&#xA;C:\Users\USER\Music\iTunes\iTunes Media&gt;cd books*&quot; It's case sensitive. And it's named `Books` on Mac. You should use something command to list all subfolders there, even the hidden ones.
Comment Sentiment = The file is POSITIVE as number of positive words are = 1


Comment = &quot;Evidently this is a bug in iBooks that ONLY manifests on non-Retina Macs. &quot; I have a Mac mini connected to a Thunderbolt display and don't suffer from this problem.
Comment Sentiment = The comment is NEGATIVE as number of negative words are = 1


Comment = (Almost 3 years later!) Yes, you can now sync ePubs (and PDFs) not purchased via the iBooks store across devices provided you have an iCloud account.
Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = 0


Comment = 2017 Update - It has been hard to find a definitive answer on the Apple Site to the following question.  Can I view DRM iBooks on my Windows PC using officially supported viewers?  It is frustrating that I buy all my DRM music and DRM Movies from iTunes but I cannot tell if I could buy DRM iBooks and view them on My PC.  Apple parses their official words carefully on their site to leave this subject ambiguous.
Comment Sentiment = The comment is NEGATIVE as number of negative words are = 3


Comment = 48 hours? So if I don't read for 48 hours it signs me out of iBooks. That insane and bizarre
Comment Sentiment = The comment is NEGATIVE as number of negative words are = 2


Comment = @AlistairMcMillan Thanks, but the solution seems to use Max OSX. Is there any way to do it on iPad? When I go to the iTunes store on iPad, I don't see any purchased book. When I go to the iBooks store on iPad, I see purchased book but no option to see format or DRM.
Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = 0


Comment = @Bravax: Do you have iBooks installed?
Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = 0


Comment = @BudAbbott The books in the iTunes books section will only sync with iBooks. Simply un-sync that section, since you don't have iBooks installed. Download, install and open 'Documents' on your device, then go to the Apps section, scroll down to the file section, select Documents and browse to your files there, then sync.
Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = 0


Comment = @Buscar웃 I use Calibre but do these categories that you add match to iBooks categories
Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = 0


Comment = @CrossRoads I think it's going to be open document in Safari, tap share or create pdf and then save it to files in *on my iPhone*. then you can find it in Files app. Also, Native pdf reader won't be a great reading experience, I suggest acrobat. Cannot speak about iBooks. Also consider having a look at the related questions in the sidebar(or bottom)
Comment Sentiment = The file is POSITIVE as number of positive words are = 2
```

Fig 3.11: Output File (Customized)



```
Comment Sentiment = The file is POSITIVE as number of positive words are = 1
Overall =    90
Comment = yea, i know that we can add pdf files to iBooks, but it wud look like a pdf and letters would be too small to read :)
Overall =    91
Comment Sentiment = The file is POSITIVE as number of positive words are = 1
Overall =    91
Comment = yes... and I just tried DropBox and it could transfer the book to iBooks... http://db.tt/XIaNlrP
Overall =    92
Comment Sentiment = The comment is NEUTRAL as number of negative and positive words are the same = 0
Overall =    92
iBooks is used positively overall    92
```

Fig 3.12: Output File Original