Lab Lecture #3.4

Calculate *tf-idf*

```
import java.io.IOException;
import java.text.DecimalFormat;

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

public class WordDocumentTfIdf
{
   public static class NewMapper extends Mapper<LongWritable, Text, Text, Text>
   {
      private static final DecimalFormat DF = new DecimalFormat("###.########");
      private static final int numberOfDocumentsInCorpus = 782;

      public void map(LongWritable key, Text value, Context context)
         throws IOException, InterruptedException
      {
         String[] wordAndCounters = value.toString().split("\t");
         String[] numbers = wordAndCounters[1].split("/");

         // Term frequency is the quotient of the number of terms
         // in document and the total number of terms in doc
         double tf = Double.valueOf(numbers[0]) / Double.valueOf(numbers[1]);

         // Inverse document frequency is the quotient between
         // the number of docs in corpus and number of docs the term appears
         double idf = (double)numberOfDocumentsInCorpus / Double.valueOf(numbers[2]);

         double tfIdf = tf * Math.log10(idf);

         context.write(new Text(wordAndCounters[0]), new Text(DF.format(tfIdf)));
       }
   }

   public static void main(String[] args) throws Exception
   {
      Configuration conf = new Configuration();
      Job job = new Job(conf, "word frequency in collection");
      job.setJarByClass(WordDocumentTfIdf.class);

      job.setOutputKeyClass(Text.class);
      job.setOutputValueClass(Text.class);

      job.setMapperClass(NewMapper.class);
      job.setReducerClass(Reducer.class);

      FileInputFormat.addInputPath(job, new Path(args[0]));
      FileOutputFormat.setOutputPath(job, new Path(args[1]));

      System.exit(job.waitForCompletion(true) ? 0 : 1);
   }
}
```