```
import java.io.IOException;
import java.util.StringTokenizer;

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.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import java.util.HashSet;
import java.util.Set;
import java.util.Map;


/*
 *       YOUSSEF NIDABRAHIM
 *   ZZ3 - F2
 *
 *       MAP/REDUCE PATTERN FOR CALCULATING MAXIMUM AND AVERAGE WORDS PER SENTENC
E
 */


public class StylePhrase {

  public static class StylePhraseMapper extends Mapper<Object, Text, Text, Custom
MaxAverageTuple>{

        private CustomMaxAverageTuple tuple = new CustomMaxAverageTuple();
        private Text sentence = new Text("Sentence");

    public void map(Object key, Text value, Context context) throws IOException,
 InterruptedException {

          StringTokenizer itr = new StringTokenizer(value.toString());
      long wordCounter = 0;
      while (itr.hasMoreTokens()) {
                        wordCounter++;
      }
      tuple.setAverage(wordCounter);
      tuple.setMax(wordCounter);
      tuple.setCount(wordCounter);

          context.write(sentence, tuple);
    }
  }

  public static class StylePhraseReducer extends Reducer<Text,CustomMaxAverageTup
le,Text,CustomMaxAverageTuple> {

        private CustomMaxAverageTuple result = new CustomMaxAverageTuple();

    public void reduce(Text key, Iterable<CustomMaxAverageTuple> values, Context
 context ) throws IOException, InterruptedException {

          int max = 0;
          int moy = 0;
          int wordCounter = 0;
          int sentenceCounter = 0;
```

```
      for (CustomMaxAverageTuple tuple : values) {
                  sentenceCounter++;
                  wordCounter = wordCounter + tuple.getCount();
                  result.setCount(wordCounter);
                  if(tuple.getMax() > result.getMax())
                        result.setMax(tuple.getMax());
      }
      result.setAverage(result.getCount()/sentenceCounter);

      context.write(key, result);
    }
  }

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

    Configuration conf = new Configuration();
    conf.set("textinputformat.record.delimiter", ".");
    Job job = Job.getInstance(conf, "Style sentences");

    job.setJarByClass(StylePhrase.class);
    job.setMapperClass(StylePhraseMapper.class);
    job.setReducerClass(StylePhraseReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(CustomMaxAverageTuple.class);

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

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