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.input.KeyValueTextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class Dictionary {

public static class WordMapper

extends Mapper<Text, Text, Text, Text>{

private Text word = new Text();

public void map(Text key, Text value, Context context

) throws IOException, InterruptedException {

StringTokenizer itr = new StringTokenizer(value.toString(),",");

while (itr.hasMoreTokens()) {

word.set(itr.nextToken());

context.write(key,word);

}

}

}

public static class AllTranslationsReducer

extends Reducer<Text,Text,Text,Text> {

private Text result = new Text();

public void reduce(Text key, Iterable<Text> values,

Context context

) throws IOException, InterruptedException {

String translations = "";

for (Text val : values) {

translations += "|" +val.toString();

}

result.set(translations);

context.write(key, result);

}

}

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "dictionary");

job.setJarByClass(Dictionary.class);

job.setMapperClass(WordMapper.class);

job.setReducerClass(AllTranslationsReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

job.setInputFormatClass(KeyValueTextInputFormat.class);

FileInputFormat.addInputPath(job, new Path("/user/bigdata/mapreduce/diccionario.txt"));

FileOutputFormat.setOutputPath(job, new Path("output"));

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

}

}