# Homework One

This is a first MapReduce homework for **undergrad students**.

**Due Date:** by email, send trigram.py to me before class on **4 March 2014**.

## What is a Trigram?

In a text file, each sequence of three consecutive words is a *trigram*. See the general case of a N-gram if you're curious on how these are used. The Google Ngram Viewer is a nice application of N-grams, for example.

## Task: Count Trigrams

The "hello world" of Hadoop (and MapReduce) is to produce a word count for text input. In that program, we get output with lines like

```
"exactly"     4151
```

indicating that the word "exactly" was found 4151 times in the input. The goal of this assignment is to count trigrams rather than words. Thus, we might expect to see a line like

```
"in a way"    7225
```

which counts the number of times the phrase "in a way" occurs in the input.

## Mrjob and Python

Use mrjob as your MapReduce framework for this assignment. See the examples, on the test machine, in these directories:

```
/opt/hadoop/mrjob/FirstMRJob
/opt/hadoop/mrjob/SecondMRJob
/opt/hadoop/mrjob/DegreeCount
```

That third directory, DegreeCount, has an example written to help you on this assignment.

## Testing

Use Tale.txt (found in FirstMRJob) for testing. You should name your program trigram.py. You can test your program with this command:

```
$ python trigram.py /opt/hadoop/mrjob/FirstMRJob/Tale.txt
```

Output, informational messages and errors will all be shown on the terminal. If you wish to save the output, try

```
$ python trigram.py /opt/hadoop/mrjob/FirstMRJob/Tale.txt >
grams.txt
```

Then, use "more grams.txt" to see the output. Suppose the output has lines like this:

```
"in a way"    439
```

For a nice, sorted output, you can do this:

```
$ sort -r -n -k 4 grams.txt | more
```

When I ran trigram.py, this was the first line of the output after sorting:

```
"said Mr. Lorry,"       50
```

(notice that Python's split does not remove punctuation, but that's OK for this small homework).

## Development

You might want to edit and do some limited testing of your trigram.py on some other platform than the test machine (using your favorite editor). On the test machine, only the *nedit* editor (along with vi and vim) is available, and even using nedit can't be done unless you have the "-Y" (or "-X") on your ssh connection.

**Note:** the only reason this assignment is not entirely trivial is that there are edge cases to deal with. Suppose a line ends with "in a" and the next line begins with "way". Then, combining these lines, we could find "in a way" as a trigram in the input. However, each call to the mapper is for one line only. That means you have to write a mapper that remembers, so to speak, information from previous calls to the mapper. The DegreeCount example shows how to do this, so looking at that example will be helpful for this assignment.