Assignment - 1

Basic Hadoop Assignment

In this Assignment you will learn how to write basic Map-Reduce programs on Hadoop. You are first supposed to perform single node installation of Hadoop on your local machine(See the Last paragraph.) and then code the following Problems. You are also given sample Input/Output files for each of the problems. The programs will be evaluated automatically on separate Test Cases so be sure to follow to input/output format. Post the queries in a separate Courses Portal Thread, if any.

1) Wordcount

The program should read text files and counts how often words occur. The input is text files and the output is text files, each line of which contains a word and the count of how often it occurred, separated by a tab.

The goal of the exercise is to count the total number of word in a document.

Approach

Understand the Data formats
Decide on the Input and output formats and implement custom classes if required
Design the Mapper
Design the Reducer

Data formats

```
this is line 1
this is line 2
……..
this is line n
```

Mapper

map() function: it just takes the input file line-by-line (in the value variable). For each line, it emits the key value (word, 1). Where word is a specific word found in that line. For example, given this line:

“This is a line”

Then the map() function will output four key-value pairs: {this, 1}, {is, 1}, {a, 1}, {line, 1}.

Reducer

All we have to do is to loop over values of the same key and sum it up.

Output

```
this  4
is  4
line  4
..
```

2) Inverted Index

An inverted index is a word-oriented mechanism for indexing a text collection in order to speed up the searching task. The inverted file structure is composed of two elements: the vocabulary and the occurrences. The vocabulary is the set of all different words in the text. For each such word a list of all the text positions where the word appears is stored. The set of all those lists is called the occurrences. These positions can refer to words or characters. Word positions (i.e., position i refers to the i-th word) simplify phrase and proximity queries.

The goal of the exercise is to build inverted index of vocabulary and their occurrences.

Data Format

```
www.kohls.com,clothes,shoes,beauty,toys
www.amazon.com,books,music,toys,ebooks,movies,computers
www.ebay.com,auctions,cars,computers,books,antiques
www.macys.com,shoes,clothes,toys,jeans,sweaters
www.kroger.com,groceries
```

Mapper

Mapper gets a line at a time, splits the line and emits key value pairs where Key is a category of product and value is the website which is selling the product. For example line retailer,category1,category2 will be emitted as (category1,retailer) and (category2,retailer).

Reducer

Reducer gets a key and a list of values, transforms the list of values to a comma delimited String and emits the key and value out.

Output

```
antiques www.ebay.com
auctions www.ebay.com
beauty www.kohls.com
books www.ebay.com,www.amazon.com
cars www.ebay.com
```

```
clothes www.kohls.com,www.macys.com
computers www.amazon.com,www.ebay.com
ebooks www.amazon.com
jeans www.macys.com
movies www.amazon.com
music www.amazon.com
shoes www.kohls.com,www.macys.com
sweaters www.macys.com
toys www.macys.com,www.amazon.com,www.kohls.com
groceries www.kroger.com
```

3) Count of Businesses in a City

Each business is located in a city. Count how many businesses are there in a city.

Data Format

Sample record:


```
{
"business_id": "rncjoVoEFUJGCUoC1JgnUA",
"full_address": "8466 W Peoria Ave\nSte 6\nPeoria, AZ 85345",
"open": true,
"categories": ["Accountants", "Professional Services", "Tax Services",
"Financial Services"],
"city": "Peoria",
"review_count": 3,
"name": "Peoria Income Tax Service",
"neighborhoods": [],
"longitude": -112.241596,
"state": "AZ",
"stars": 5.0,
"latitude": 33.581867000000003,
"type": "business"
}
{
"business_id": "G2Re2E5Jkv_UF1xrenSsDg",
"full_address": "4500 E Cactus Rd\nPhoenix, AZ 85032",
"open": true, "categories": ["Department Stores", "Fashion", "Shopping"],
"city": "Phoenix",
"review_count": 11,
"name": "Macy's",
"neighborhoods": [],
"longitude": -111.9820907,
"state": "AZ",
"stars": 4.0,
"latitude": 33.599323599999998,
"type": "business"
}
```


Parsing

The data is in JSON format. It is well structured for processing. However we cannot use FileInputFormat because we need to strip down the braces and there could be nested JSON structure inside.

Hence it is clear we have to write our own custom Input Format class. Fortunately, the json library comes with a parse, which we can directly use.
Since the data is JSON and we need to parse the JSON to get the key and value pairs. A custom InputFormat is written, which will process one JSON structure at a time. In order to do this, we need to read one JSON line at a time and convert into to key/value pairs for processing. This is the job of the RecordReader, which will have the necessary intelligence built in to read one JSON structure at a time.

Mapper

The Mapper is simpler class similar to the classic wordcount program in the examples. We are counting the businesses in a city instead of the wordcount.

Reducer

The Reducer is also a simple one similar to the classic wordcount example. In this case, we are counting the total businesses in the city.

Output

```
Peoria 1
Phonix 1
```

4) Business Reviews by Month and Season

Count the reviews by months and season.

Data format


```
{
"votes":
{"funny": 0, "useful": 5, "cool": 2},
"user_id": "rLtl8ZkDX5vH5nAx9C3q5Q",
"review_id": "fWKvX83p0-ka4JS3dc6E5A",
"stars": 5,
"date": "2011-01-26",
"text": "My wife took me here on my birthday for breakfast and it was excellent. The weather was perfect which made sitting outside overlooking their grounds an absolute pleasure. Our waitress was excellent and our food arrived quickly on the semi-busy Saturday morning. It looked like the place fills up pretty quickly so the earlier you get here the better.\n\nDo yourself a favor and get their Bloody Mary. It was phenomenal and simply th best I've ever had. I'm pretty sure they only use ingredients from their garden and blend them fresh when you order it. It was amazing.\n\nWhile EVERYTHING on the menu looks excellent, I had the white truffle scrambled eggs vegetable skillet and it was tasty and delicious. It came with 2 piece of their griddled bread with was amazing and it absolutely made the meal complete. It was the best \"toast\" I've ever had.\n\nAnyway, I can't wait to go back!",
"type": "review",
"business_id": "rncjoVoEFUJGCUoC1JgnUA"
}
```


Mapper

```
// parse 2011-01-26, to get the month and season, like 01 refers Jan month and Winter ( winter - nov, dec, jan, feb. spring - march, april, may, june, july. monsoon - august, sept, oct)
```

Mapper outputs the months and its count and also season and its count.

Output for map:

june 1
july 1
august 1
june 1
spring 1
spring 1
monsoon 1
spring 1

Reducer

The Reducer is similar to the WordCount. It simply counts the count of reviews by month and seasons.

Output

june 2
july 1
august 1
spring 3
monsoon 1

Important Guidelines:

- Deadline is on 11th Sept, 9pm. (will not be extended in any case)
- **Plagiarism of any kind (from others or from internet) will be detected, and will be awarded a straight 0 in the assignment.**
- There will be an automated evaluation in our hadoop cluster, and a manual evaluation in case of some problem.
- You can only code in Java as hadoop is implemented in Java. There are libraries available in Python/Ruby but they make the computation slow. So, you are allowed to code only in Java.
- To install single node hadoop cluster on your laptop, follow each and every step: http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/