Analyzing Twitter Data using MapReduce

Chan-Ching Hsu

Purpose

Use Hadoop MapReduce for analyzing data from Twitter. The data from twitter consists of a number of messages of “tweets”. Each tweet, in addition to text, can also contain a “hashtag”, which can be thought of as a topic to which a tweet can belong. In addition, a tweet from one user can be “retweeted” by another user and this can also be inferred by looking at fields within the tweet.

Dataset

The data is in the JSON format.

JSON

JSON is in its most simple explanation, a lightweight data-interchange format. It is commonly used in web applications to package data.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most language, this is realized as an *array*, vector, list, or sequence.

An object is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by ‘,’

Example of JSON

```
[
    {
        “name”: “John Smith”,
        “salary”: 1000,
        “hobby”: “Bob-sledding”
    },
    {
        “name”: “Paul Bunyan”
        “salary: 13200,
        “hobby”: “Cricket”
    }.
    {
        “name”: “Daisy”,
        "salary": 10000,
        "hobby": "Hockey"
    }
]
```

This is an array of objects with each object having an attribute of "name", "salary", and "hobby". A more complete explanation at http://json.org/

There are many third part JSON parsers.

- GSON: https://github.com/google/gson
- Simple-json: https://code.google.com/archive/p/json-simple

Use the following command to add external jars:

```
Hadoop jar /path/to/jar Class –libjars LIBJARS
```

The input format specifies how we want to cut our data into **splits**. Each **split** is sent to a different mapper. Once in a mapper the record reader breaks the **split** into **records**. A split has many records in it. Each record is a single <key, value> pair.

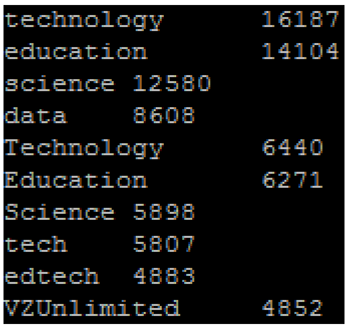
We can use the record reader to read in a single JSON object. Once done, it will read through a split until it finds the end of the JSON object. To do this we need to define the bounds of a JSON object. In JSON, each object starts with a '{' and ends with a '}'. Therefore, a good approach for writing code to recognize a JSON object would be to read starting at a '{' and reading until we find the matching '}'. We can write this new logic in a custom record reader. The concepts and some example code are explained well here: https://hadoopi.wordpress.com/2013/05/31/custom-recordreader-processing-string-pattern-delimited-records/

For our record reader we need to implement 6 different methods:
Initialize(): Prepares the record reader to start reading
nextKeyValue(): Creates the next Key Value pair. It will return true until it reaches the end of the file
getCurrentKey(): Returns the current key
getCurrentValue(): Returns the current value
getProgress(): Reports how far into the file we have read
close(): Used by the framework for cleanup

Experiment 1

Find the top ten most common hashtags. If a hashtag appears twice in the same tweet, only count the hashtag once. Many tweets will have the same text because one user retweeted the tweet. This does not count as a duplicate.

Output format: hashtag/count

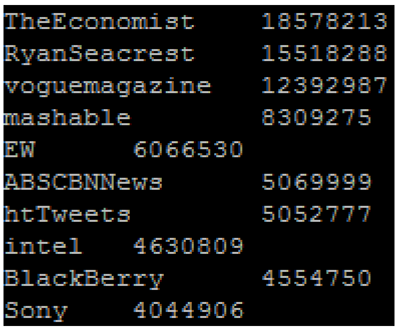


```
technology      16187
education       14104
science 12580
data    8608
Technology      6440
Education       6271
Science 5898
tech    5807
edtech  4883
VZUnlimited     4852
```

Experiment 2

Find the top ten most followed tweeters. Output the name and number of followers for each tweeter. A user field describes the tweeter in more detail. There is a “screen_name” field and a “followr_count” field. A user can have different follower counts across different tweets. Choose the highest occurring follower count for each tweeter.

Output format: tweeter/followers count

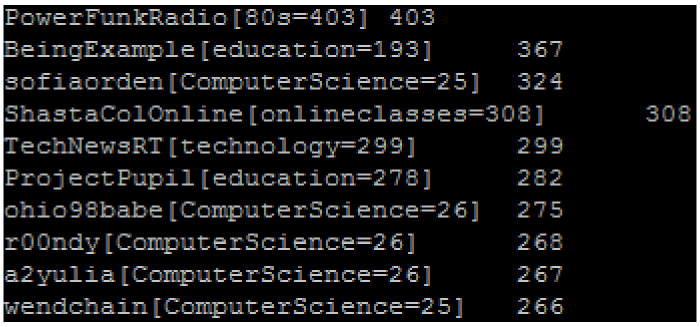


```
TheEconomist    18578213
RyanSeacrest    15518288
voguemagazine   12392987
mashable        8309275
EW      6066530
ABSCBNNews      5069999
htTweets        5052777
intel   4630809
BlackBerry      4554750
Sony    4044906
```

Experiment 3

For each of the tweeters in the top ten most prolific tweeters find the most commonly used hashtag. Prolific means the most number of posts, not the most number of followers. We want to see what these prolific tweeters talk about the most. This experiment will be more difficult as you will need to keep tweet information together.

Output format: tweeter/[most used hashtag=hashtag count]/number of posts



```
PowerFunkRadio[80s=403] 403
BeingExample[education=193]      367
sofiaorden[ComputerScience=25]  324
ShastaColOnline[onlineclasses=308]       308
TechNewsRT[technology=299]       299
ProjectPupil[education=278]      282
ohio98babe[ComputerScience=26]  275
r00ndy[ComputerScience=26]       268
a2yulia[ComputerScience=26]      267
wendchain[ComputerScience=25]    266
```