OpenStreetMap Project Data Wrangling with MongoDB

Anav Gupta

Map Area : Birmingham, England, UK

https://mapzen.com/data/metro-extracts/metro/birmingham_england/

1. Data Overview

This section contain some of the statistics about the chosen data-set.

- File Size
  - 98 MB (When compressed using Bzip2)
  - 1.5 GB (Uncompressed)

- No of Unique Users in the Original OSM file
  - Using *users.py* File
  - 2370

- No of Tags in the Original OSM File
  - Using *mapparser.py* File
  - 'node': 6361009
  - 'relation': 7122
  - 'way': 1067779

Further Statistics are computed after the file is cleaned and imported into the MongoDB database.

2. Problems Encountered in the Map

After downloading the birmingham_england.osm metro extract from the MapZen website. I create a sample of the birmingham_england.osm file. This file contain only 1 / 10 of all nodes present in the original OSM file. After doing first rounds of the auditing, I found many discrepancies in the sample OSM file.

- Inconsistent Street Names. e.g: Ave, Aveune for Avenue.
- Inconsistent House Numbers.
- Inconsistent Phone Numbers e.g: "+44 121 632 5602", "01455 213838 ext 20" // To-do for Later

Inconsistent Street Names

After going through the auditing process for the sample OSM file, I found that there are some discrepancies in the name of the streets. To make sure that I find all such kind of typos in the streets

names, I can auditing against the original OSM file. I routed the results into a file. Keeping the results in a file made it possible for me to go through all the mistakes or typos in the street names. It made it possible for me to make a list of all the instances of the mistake so that they can be rectified. E.g: In some instances the name 'Avenue' was written in short hand as 'Ave'. In some instances there were mistakes in the spellings of the 'Avenue'.

Inconsistent House Numbers

Auditing process for the house numbers showed me some inconsistencies in how multiple house numbers were separated. A comma, semicolon or an ampersand sign were used on multiple occasions. There were some errors present as well. E.g: 'Unit12'. Some short hand usage e.g for Apartments, 'Apt' was used.

Auditing House Numbers seemed particularly more difficult. There was a lot of data that I didn't understood. The meaning of 1A/1B, 34-45,5 or 8?, 2B/C, 'rear of 100', 'Unit D & E', 129/a and like terms seemed to baffled me. Then I chose to ask the UK community of the Reddit of their significance. The response was phenomenal and it helped me to understand what kind of changes I could make in the house numbers.

Street names and House numbers were firstly separately audited and their results were studied. The erroneous entries, the ones that could be systematically cleaned were picked. They were amassed into a dictionary for the mapping purpose. This was all done so to remove the errors.

Insufficient Processing Capacity

Auditing of 1.5 GB file is a mammoth task. It requires that you have a system with high processing power and high memory (RAM). My system was an average on both of the scales. So even after making sure that my code is not going failed, it failed multiple times. I was not able fully process the file. The processing used to stop somewhere in the middle. I thought my system had hanged up or have thrown up some another issue. This continued until I noticed a trend in this. Whenever I writing to the JSON file, the processing used to stop when file becomes 852 MB.

This mean that after writing 852 MB data on the file, the processing used to stop. Mainly because my system didn't have memory required to continue further. So I had to find a way to bypass this and I did. I first imported the 852 MB file that I had into MongoDB. Then I calculated the number of lines the file contained. I did it by calculating the no of document that had been imported into the MongoDB. It came out to be 3887348.

Instead of writing a new file, I thought to append to the existing file. I created a check in the data.py file to skip all the documents before this number. I was successful in my trial. It worked and was able to process the file further to make it up to 1.5 GB. But this was not it. After reaching this limit, the processing seemed to stuck in the similar fashion. I had to repeat the earlier process of importing the file into MongoDB and again calculating the no of line in the file. This time it came out to be 6943441 no of lines.

I updated the data.py file for this check. After this when I again started the processing of the Original OSM file, I was able to reach to the end of the file and now I knew that my Original OSM file has been fully processed.

3. Some Queries in the MongoDB database

Creating an Index in the Database

```
db.birmingham.createIndex({'type' : 1, 'id' : 1}, {'unique' : true})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

Since My database is of more than 75 Lakhs documents, having an index is worth having.

Calculating Top 11 Amenities

```
db.birmingham.aggregate([{'$match' : {'amenity': {'$ne' : null}}}, {'$group' : {'_id' : '$amenity',
'count' : {'$sum' : 1}}}, {'$sort' : {'count' : -1}}, {'$limit' : 11}])
{ "_id" : "parking", "count" : 7338 }
{ "_id" : "post_box", "count" : 3448 }
{ "_id" : "pub", "count" : 2459 }
{ "_id" : "school", "count" : 2051 }
{ "_id" : "place_of_worship", "count" : 2007 }
{ "_id" : "fast_food", "count" : 1983 }
{ "_id" : "bench", "count" : 1605 }
{ "_id" : "telephone", "count" : 1236 }
{ "_id" : "restaurant", "count" : 1176 }
{ "_id" : "cafe", "count" : 874 }
{ "_id" : "bicycle_parking", "count" : 854 }
```

Parking seems to be on top of all amenities. Seems plausible with all the increase in the number of Motor Vehicles. Parking is a necessity.

Finding the count of Places of Worship based on the Religion

```
db.birmingham.aggregate([{'$match' :{'amenity' : 'place_of_worship'}}, {'$group' : {'_id' :
'$religion', 'count' : {'$sum' : 1}}}, {'$sort' : {'count' : -1}}, {'$limit' : 11}])
{ "_id" : "christian", "count" : 1645 }
{ "_id" : null, "count" : 195 }
{ "_id" : "muslim", "count" : 96 }
{ "_id" : "sikh", "count" : 39 }
{ "_id" : "hindu", "count" : 16 }
{ "_id" : "buddhist", "count" : 7 }
```

```
{ "_id" : "jewish", "count" : 4 }
{ "_id" : "spiritualist", "count" : 2 }
{ "_id" : "multifaith", "count" : 2 }
{ "_id" : "scientologist", "count" : 1 }
```

Christian seem to be the religion with most places of worship, which is expected in a city in England. What is more interesting that the city have people with variety of religions.

Finding the type of denominations present

```
db.birmingham.aggregate([{'$match' :{'amenity' : 'place_of_worship', 'religion' : 'christian'}},
{'$group' : {'_id' : {'religion' : '$religion', 'denomination' : '$denomination'}, 'count' : {'$sum'
: 1}}}, {'$sort' : {'count' : -1}}, {'$limit' : 11}])
{ "_id" : { "religion" : "christian" }, "count" : 666 }
{ "_id" : { "religion" : "christian", "denomination" : "anglican" }, "count" : 443 }
{ "_id" : { "religion" : "christian", "denomination" : "methodist" }, "count" : 158 }
{ "_id" : { "religion" : "christian", "denomination" : "catholic" }, "count" : 114 }
{ "_id" : { "religion" : "christian", "denomination" : "baptist" }, "count" : 90 }
{ "_id" : { "religion" : "christian", "denomination" : "roman_catholic" }, "count" : 26 }
{ "_id" : { "religion" : "christian", "denomination" : "evangelical" }, "count" : 19 }
{ "_id" : { "religion" : "christian", "denomination" : "jehovahs_witness" }, "count" : 16 }
{ "_id" : { "religion" : "christian", "denomination" : "pentecostal" }, "count" : 15 }
{ "_id" : { "religion" : "christian", "denomination" : "quaker" }, "count" : 13 }
{ "_id" : { "religion" : "christian", "denomination" : "mormon" }, "count" : 10 }
{ "_id" : { "religion" : "christian", "denomination" : "united_reformed" }, "count" : 10 }
```

Of all the Christian churches present in the area, the *'anglican'* denomination seems to outnumber the rest of them.

Finding the Top 10 Sources of the Data

```
db.birmingham.aggregate([{'$group' : {'_id' : '$source', 'Count' : {'$sum' : 1}}}, {'$sort' :
{'Count' : -1}}, {'$limit' : 10} ])
{ "_id" : null, "Count" : 6568131 }
{ "_id" : "bing", "Count" : 461059 }
{ "_id" : "bcc_dec_2016", "Count" : 42974 }
{ "_id" : "OS_OpenData_StreetView", "Count" : 42268 }
{ "_id" : "survey", "Count" : 42159 }
{ "_id" : "bing, local knowledge", "Count" : 36158 }
{ "_id" : "Warwicks CC Aerial Imagery 2013", "Count" : 31813 }
{ "_id" : "Bing", "Count" : 31197 }
{ "_id" : "WMCA", "Count" : 12574 }
{ "_id" : "bing,local knowledge", "Count" : 12262 }
```

Whenever we are playing with Data, the source of the data is of utmost important. We must be sure of the credibility of the source. As you can see in the top ten most sources for the data, there are many discrepancies. 'Bing' appears in four rows in the top ten sources. This shows that there can be some improvement in the naming of the sources. Keeping this for LATER.

Finding the Top 10 Contributors.

```
db.birmingham.aggregate([{'$group' : {'_id' : '$created.user', 'Count' : {'$sum' : 1}}}, {'$sort' :
{'Count' : -1}}])
{ "_id" : "brianboru", "Count" : 3874107 }
{ "_id" : "blackadder", "Count" : 512029 }
{ "_id" : "Miked29", "Count" : 468104 }
{ "_id" : "mrpacmanmap", "Count" : 184569 }
{ "_id" : "Curran1980", "Count" : 156185 }
{ "_id" : "James Derrick", "Count" : 134831 }
{ "_id" : "PeterP", "Count" : 130836 }
{ "_id" : "srbrook", "Count" : 113804 }
{ "_id" : "richardwest", "Count" : 109339 }
{ "_id" : "The Maarssen Mapper", "Count" : 97394 }
```

The Open Street Map works because of the users who have contributed to the formation of the maps. We must thanks them for helping us, the global community to provide us the Open Source Map data.

Finding the Top most Cuisines served in the Restaurants

```
db.birmingham.aggregate([{'$match' : {'amenity' : 'restaurant'}}, {'$group' : {'_id' : '$cuisine',
'Count' : {'$sum' : 1}}}, {'$sort' : {'Count' : -1}}])
{ "_id" : null, "Count" : 705 }
{ "_id" : "indian", "Count" : 178 }
{ "_id" : "chinese", "Count" : 61 }
{ "_id" : "italian", "Count" : 50 }
{ "_id" : "thai", "Count" : 17 }
{ "_id" : "french", "Count" : 13 }
{ "_id" : "pizza", "Count" : 11 }
{ "_id" : "american", "Count" : 11 }
{ "_id" : "regional", "Count" : 9 }
{ "_id" : "chicken", "Count" : 8 }
```

It's been said that people are known by the food they like and eat. This is very interesting to see that Indian Cuisine is the most served Cuisine. It can mean very different things. For E.g :

1. There is a large population who like Indian cuisines.
2. There is a large number of Indian peoples living in the area.
3. Some data is missing.

Find the no of Contributors

```
db.birmingham.aggregate([{'$group' : {'_id' : '$created.uid' , 'count' : {'$sum' : 1}}}, {'$count'
:'No. of Contributors'}])
{ "No. of Contributors" : 2351 }
```

This is odd. We have previously calculated that there were 2370 Unique users who have contributed to the development of the map. The data in the MongoDB shows that there are only 2351 Unique

users who have contributed. This difference can be explained that using following of the propositions.

1. We are only importing 'node' and 'way' into our MongoDB database. So there are users who have not contributed in them.
2. While cleaning the file and creating the JSON file we may have lost some of the data.

There will definitely be some more appropriate explanation for this difference in the number of the users.

4. Improvements

1. Sources: By improving the source of the information we will be more equipped to tackle the problem of data Wrangling.

    - Benefits:
        1. We will be in a better position to verify the validity of the data.
        2. We can focus quickly on the wrangling process, if we are already sure about the validity of the source and the data provided.

    - Anticipated Issues:
        1. We can never be sure of the validity of the data provided, even if it is from a valid and a vouched source.
        2. Everybody is susceptible to making errors. Errors that are unintentional are acceptable, but errors were intentional and carefully planned there. We must be clinical about the data we have in our hand.

2. Standardization: I know that I am dealing with the Open Street Map data, which is an open sourced Data base. People from their region contribute in the development of the the maps. I think the way the data is collected must be improved. Through out my project I struggled to understand the data that is typical to a place. If we can create a documents, a meta data document that could describe all the data that we store, the analysis process will become smoother. For eg: A document to show Postal code is formed and how it is used in the real world. I highly recommend this. This has the power to fuel some more advancements in the data.

    - Benefits:
        1. A major portion of any data wrangling task is to first fully understand the data under consideration. Having a Meta Document which can describe the data can effectively reduce the time in understanding the data.
        2. With more standardization of the data, we can conform to a similar kind of information through out the world.

    - Anticipated Issues:
        1. A world is a collection of peoples from different Region, Religion, Cultures. Each Community have their own ways to describe the information around them. It is sufficiently difficult to get them to concur on the standardized way of information.

2. Information is propagated through the people using it. Over the ages the people have changed the way information is sent out, to make it more comfortable for the people using that information. We like to amends of our own to meet our needs.

5. Conclusion

This was a great activity to learn the process of data wrangling. I really understood why is it important to first understand the data that you are working on. Until you really understand the data you won't be able clean, make assumptions or even work with the data. I understand that we need someone to vouch for the data that we collect, because until we have someone with the credibility, we can't be sure of the truthiness of the data.