Project 3 : Data Wrangling with MongoDB

Ashutosh Singh

Map Area: Mumbai, Maharashtra, India
OSM File from mapzen.com

1. Introduction and Motivation

I chose the Mumbai area as I am residing here for the past 3 years and wanted to explore the city. Also I wanted to look at the data quality as not many of my colleagues have heard of Open Street Maps and not many people here are educated that we can update the maps and improve the data quality (My personal perception before starting this project).

The Structure of the code is as follows.

audit.py | This contains the auditing functions like finding the inconsistent or incorrect data and updating it according to norms.

data.py | It is used to parse the **OSM** file and uses the **audit.py** to correct the data found and then write the data to a json file.

The data from json file is imported to mongoDB.

queries.py | The queries from the **Additional Section** is written in this file.

2. Problems Encountered in the maps

Street Names

Mumbai is not a planned city. Also due to diversity of cultures in the city the names of places are not consistent as in most first world cities. Sometimes the names are in English, but in many instances the street name is in hindi and is just transliterated in English. Following problems are visible from a glance at data

- Various names in the end of street eg. ***marg,road,Rd),path,wadi, gali, chawl, chowk***
- Multiple names of same street within parenthesis eg. ***Maratha mandir marg (Club road)***
- Non string data eg ***4620;,http://mha.gov.in***

First the problem characters were checked and if found the string is removed from being added as clean data. Then everything inside parenthesis is removed to remove duplicate street names. Then the abbreviated names were replaced with the full names and minor incorrect spelled names were also corrected. Finally all the names were changed to Title Case.

Postal Codes

In India, we have 6 digit postal codes. In each local area the first few digits remain constant and the last 2/3 digits change in that area. For Mumbai the pattern is **400XXX** hence only the last 3 digits will change. When going through the sample data following problems occured.

- Whitespaces in postal code eg. ***400 099***
- Postal codes not starting with 400 eg. ***402075***, ***55035***
- Postal codes not having six digits eg. ***66, 49***

The identification of these were done first by using a regular expression and then replacing them by either correcting or adding extra digits to them / removing whitespaces or removing them from the data.

3. Overview of the Data

This section contains the basic statistics about the dataset and the queries used to fetch them

File Sizes

mumbai_india.osm : 335773 kB

mumbai_india.json : 395533 kB

Number of documents :

```
db.mumbai.find().count()
```

1880457

Number of nodes :

```
db.mumbai.find({type:”node”}).count()
```
1674702

Number of ways :

```
db.mumbai.find({type:{“way”}).count()
```
205554

Number of unique users

```
db.mumbai.distinct(“created.user”).length
```
1086

Top Contributing user

```
db.mumbai.aggregate( [
{"$group": { "_id": "$created.user" , "count": {"$sum": 1}}},
{"$sort": { "count" : -1}},
{"$limit": 1}
] )
```
[{u'count': 71060, u'_id': u'parambyte'}]

Users with single edit

```
db.mumbai.aggrefate([
{"$group" : { "_id": "$created.user", "count": { "$sum": 1}}},
{"$group" : { "_id": "$count" , "num_users": { "$sum" : 1}}},
{"$sort" : {"_id" : 1}},
{"$limit" : 1}
] )
```
[{u'num_users': 205, u'_id': 1}]

Total number of buildings

```
db.mumbai.find({"building" : "yes"}).length
```
157595

4. Additional Analysis

Places of Worship / Religion

India is a land of religions,languages and cultures. Lets take a look at the places of worship in the city as sorted order.

```
pipeline = [
            {"$match" : {"amenity" : {"$exists" : 1}, "amenity" : "place_of_worship"}},
            {"$group" : {"_id" : "$religion" , "count" : {"$sum" : 1}}},
            {"$sort" : {"count" : -1}}
        ]

pprint( list(db.mumbai.aggregate(pipeline)))
```

[{u'_id': u'hindu', u'count': 128},
{u'_id': u'muslim', u'count': 97},
{u'_id': u'christian', u'count': 58},
{u'_id': None, u'count': 51},
{u'_id': u'buddhist', u'count': 14},
{u'_id': u'jain', u'count': 9},
{u'_id': u'zoroastrian', u'count': 9},
{u'_id': u'sikh', u'count': 6},
{u'_id': u'jewish', u'count': 4},
{u'_id': u'hare_krishna', u'count': 1},
{u'_id': u'sikhs', u'count': 1}]

As found, there are 7 religions place of worship found. The most pupular religion is hindu and then muslim also represented in the india's demographic data. Now few places of worship don't have a religion field in them. Printing them we get

```
pipeline = [
            {"$match" : { "amenity" : {"$exists" : 1 } , "amenity" : "place_of_worship"
                         ,"religion":None}},
            {"$project" : {"_id" : 0, "name" :1}},
            {"$limit" : 5 }
        ]

    pprint( list(db.mumbai.aggregate(pipeline)))
```

[{u'name': u'PANCHAMUKHI SRI HANUMAN MANDIR'},
{u'name': u'Nuri Baba Darga'},
{u'name': u'Saibaba Mandir'},
{u'name': u'Shiv Temple'},
{u'name': u'Don Bosco Church'}]

In the top 5 items we have 1 church, 3 temples and a dargah(muslim place of worship) So it looks like the entries for these are incomplete and can be done by editing each place manually

Top 10 Amenities

First take a look at the top 10 amenities

```
pipeline = [
            {"$match" : {"amenity" : {"$exists":1}}},
            {"$group" : {"_id"  : "$amenity" , "count" :{"$sum" :1}}},
            {"$sort" : {"count" : -1}},
            {"$limit" : 10}
        ]

pprint( list(db.mumbai.aggregate(pipeline)))
```

[{u'_id': u'place_of_worship', u'count': 378},
{u'_id': u'restaurant', u'count': 267},
{u'_id': u'school', u'count': 244},
{u'_id': u'bank', u'count': 228},
{u'_id': u'hospital', u'count': 150},
{u'_id': u'fuel', u'count': 124},
{u'_id': u'parking', u'count': 122},
{u'_id': u'bus_station', u'count': 114},
{u'_id': u'cafe', u'count': 114},
{u'_id': u'college', u'count': 96}]

Well, it looks like we have more places of worship than we have schools and hospitals. There may be a bias here that schools are marked by the local users whereas we know of places of worship which are far away. So many users know of religious places than they know of schools and hospitals.

Most active year (editing)

Looking at the time when most of the editing is done

```
pipeline = [
           { "$group" : { "_id" :
                   {"year" : "$created.year" ,
                    "month" :"$created.month"
                   },
                    "editCount" : {"$sum":1}}
           },
           { "$group" : { "_id" : "$_id.year",
                          "month" : {
                                  "$push" : {
                                          "month" : "$_id.month",
                                          "edits" : "$editCount"
                                     },
                                  },
                          "count": {"$sum": "$editCount"}
                        }},
           {"$sort" : {"count" : -1} },
           {"$limit" : 1 }

pprint( list(db.mumbai.aggregate(pipeline)))
```

[{u'_id': u'2015',
u'count': 1359350,
u'month': [{u'edits': 5701, u'month': u'05'},
{u'edits': 2174, u'month': u'02'},
{u'edits': 371148, u'month': u'07'},
{u'edits': 3131, u'month': u'01'},
{u'edits': 4169, u'month': u'03'},
{u'edits': 3516, u'month': u'04'},
{u'edits': 969511, u'month': u'06'}]}]

2015 is the year when most of the editing is done. This may be because now users in India are getting more tech savvy and updating the data.

For a final query lets find the month in which most of the editing is done

```
pipeline = [
           {"$group" :{ "_id" : "$created.month" , "count" : {"$sum" :1}}},
           {"$sort" : {"count" : -1}},
           {"$limit":2}
       ]
pprint( list(db.mumbai.aggregate(pipeline)))
```

[{u'_id': u'06', u'count': 978172},
{u'_id': u'07', u'count': 406501}]

It's odd but the most active month for editing are **June** and **July**

5. Other Ideas about the dataset

The data quality in the open street maps is widely varying depending upon the regions and also how much local volunteers are interested in improving it. Like for a big city in US or Europe there are enough people with the motivation and the skills to update an area with relevant details. But in a country like India there are few skilled volunteers to take up the task.

Also in India, standardization takes a second place when put together with locality. Even thought the volunteers are updating a map they may put it in commonly used words than taking a standardized approach. There are 2-3 approaches to bring more volunteers and the get cleaned data.

a. **Gamification** : The concept of rewarding people with virtual badges, points, karma is the cornerstone of many communities like *foursquare, reddit, stackoverflow etc.* Prominent display of user statistics on the osm website and also on the searched maps will prompt more users to contribute.

b. **High Quality tools** : If high quality tools are available to the community users to easily edit and mark the maps (like Google Map Maker). Also the inbuilt quality checks can improve the quality of data to a new level.

c. **Incorporating more data** : The users at local level should incorporate more data from the local administrative authorities to cross-validate the present data and add more info to it.

6. Conclusion

India is a land of a vast array languages and communities.The data also conforms to it. There are various names of streets and buildings. The data is also filled with the how the names/zipcodes are used in daily life instead of a common standard. If proper checks are implemented then some the data entered can be adhered to the standard but again in this type of data (maps) localization is important. We need more active volunteers to clean and maintain the data just like in wikipedia. I think the data will improve over time as people get more used to Open Street Maps.

7. References

- OSM XML wiki
- Mapzen Metro Extracts
- Python Regex
- Elementree IterParse
- mongoDB Aggregation
- mongoDB Operators