OpenStreetMap Project
Data Wrangling with MongoDB

Pratik Asarkar

Map Area: Pune, Maharashtra, India

Used overpass API with the co-ordinates from following link:
http://www.openstreetmap.org/export#map=14/18.5203/73.8543

1. Problems Encountered in the Map

After downloading the dataset and running it against some code in python, I noticed the following problems in the data and I will discuss it in the following order:

- Non-English user names
- Abbreviated fuel brands
- Inconsistency in country, state,city names and postal codes.

Non-English user names

After exploring the dataset, it came to notice that there existed some non-english user names in the dataset. The number of non english names were handful which made it possible for me to convert them to english using google translate. Then I replaced those names with the translated names using python.
Eg : There exists a user with marathi name 'शंतनू'. Its English translation is 'Shantanu'. So the data stored in the mongodb database will be the translated english version.

Abbreviated fuel brands

Once the data was imported to MongoDB, some basic querying revealed fuel brand abbreviations. I updated all strings in problematic fuel brands, such that "IOCL" becomes "Indian Oil Corporation Limited".

Inconsistency in country, state,city names and postal codes.
By looking at the dataset, it was clear that the “addr:country” tag had abbreviated country code as its value. For clear understanding the country code was changed from “IN” to “India”.
The “addr:state” tag had inconsistent values like “MH”,”Maharashtra”,”maharashtra”. These were changed to “Maharashtra” as all the values represent the same state making it more consistent for querying.
The “addr:city” also had inconsistent values like “Pune”,”pune”,”Magarpatta City”. As the dataset we are working with belongs to Pune city, The value of city has been changed to Pune to avoid any data inconsistencies.
Some postal codes had white spaces in them. So I got rid of those inconsistencies too.

2. Data Overview

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

File sizes

interpreter.osm ..... 52.67 MB
interpreter.osm.json .... 60.12 MB

Number of documents

```
> db.project3.find().count()
277255
```

Number of nodes

```
> db.project3.find({"type":"node"}).count()
232788
```

Number of ways

```
> db.project3.find({"type":"way"}).count()
44467
```

Number of unique users

```
> db.project3.distinct("created.user").length
```

298

Top 3 contributing user

```
db.project3.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":3}])
{ "_id" : "singleton", "count" : 32300 }
{ "_id" : "vamshikrishna", "count" : 19332 }
{ "_id" : "kranthikumar", "count" : 17269 }
```

Number of users appearing only once (having 1 post)

```
> db.project3.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}}, {"$group":{"_id":"$count", "num_users":{"$sum":1}}},
{"$sort":{"_id":1}}, {"$limit":1}])
{ "_id" : 1, "num_users" : 88 }
# “_id” represents postcount
```

3. Additional Ideas

Top user contribution percentage (“singleton”) - 11.65%
Combined top 3 users' contribution (“singleton”,“vamshikrishna” and “kranthikumar”) - 24.85%
Combined Top 10 users contribution - 53.14%
Combined number of users making up only 1% of posts - 249 (about 83% of all users)
Regional Language - Marathi (About 52% of all tags with names have marathi names which gives a clear indication that a major population in the region is familiar with marathi language)

After taking a look at the dataset, I realized that there could be some more data added to the locations in the dataset. For example, we could add an extra tag with the distance of that location from the major stops like the Railway Station, Bus Stand, Airport, etc.
For implementing the above idea, we could use comma separated values in the “v” attribute of the tag. Each value position could represent the distance of the location from a particular location.

The benefit with the above idea is that it could help the user get the additional information when hovered upon the location and do not have to explicitly find the distance.
But the limitation for implementing such a technique is that there would be requirement of having an accurate data available with us about the distance of major locations(Bus stand, Airport, etc.) from a particular location. Also collecting and managing so much data for every possible location could be a tedious job. The data could be made available from some third party apis but there could also be some cost associated with it which has to be beared.

Additional data exploration using MongoDB queries

Top 10 appearing amenities

```
> db.project3.aggregate([{"$match":{"amenity":{"$exists":1}}},
{"$group":{"_id":"$amenity", "count":{"$sum":1}}}, {"$sort":{"count":-1}},
{"$limit":10}])
{ "_id" : "restaurant", "count" : 122 }
{ "_id" : "school", "count" : 87 }
{ "_id" : "place_of_worship", "count" : 86 }
{ "_id" : "bank", "count" : 63 }
{ "_id" : "atm", "count" : 45 }
{ "_id" : "cafe", "count" : 42 }
{ "_id" : "fuel", "count" : 35 }
{ "_id" : "hospital", "count" : 35 }
{ "_id" : "parking", "count" : 34 }
{ "_id" : "college", "count" : 33 }
```

Biggest religion

```
>db.project3.aggregate([{"$match":{"amenity":{"$exists":1},"amenity":"place_of
_worship"}},{"$group":{"_id":"$religion",
"count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":1}])
{ "_id" : "hindu", "count" : 57 }
```

Most popular cuisines

```
>db.project3.aggregate([{"$match":{"amenity":{"$exists":1},"amenity":"restaura
nt"}}, {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},{"$sort":{"count":-1}},
{"$limit":5}])
```

```
{ "_id" : null, "count" : 69 }
{ "_id" : "indian", "count" : 23 }
{ "_id" : "regional", "count" : 5 }
{ "_id" : "pizza", "count" : 5 }
{ "_id" : "vegetarian", "count" : 4 }
```

Villages nearby

```
>db.project3.aggregate([{"$match":{"place":{"$exists":1},"place":"village"}},{"$p
roject": {"_id":"$name"}}]).pretty()
{ "_id" : "Theur" }
{ "_id" : "Pedgaon" }
{ "_id" : "Babhulsar Budruk" }
{ "_id" : "Nanvij" }
{ "_id" : "Kuldharan" }
{ "_id" : "Peth" }
```