OpenStreetMap Sample Project
Data Wrangling with MongoDB

Dalia Tarek

Map Area: Cairo, Egypt
https://mapzen.com/metro-extracts/

File size: 69.9 MB

1. Overview of the map file

After downloading the OSM file of the selected area I run some scripts to get an overview of the data.

1. Tags overview

```
bounds: 1
member: 824
nd: 413143
node: 320879
osm: 1
relation: 127
tag: 123562
way: 44197
```

2. Number of users have contributed to this file

518 unique users

2. Problems Encountered in the Map

1. Auditing Street names of nodes and ways

I began auditing the street names of nodes, ways, ways' names and ways' english names to find any pattern for writing it but there was no specific pattern.

Here is a sample of the auditing results

'Ahmed Emara; Ahmed Emara; Ahmed Al-Awamry; Maher’,
'Ahmed Zaki Street’,
'Hamad Yassin st.’,
'Alex-Cairo Desert Rd’,
'Othman ibn Affan Sq.', 'Victoria Sq.
'13 St
St. Josef Church for Franciscan Padres’
'Misr El Nour Road’,
'Road 10’,

As you can see, there is no standard format for writing it, some have a type others have type abbreviations and others have no type, the type itself has no fixed place to be written it sometimes in the beginning, last, and sometimes in between.

So, we can improve this part of data as following.

The usage of abbreviations, and its position in the name

I worked on detecting the abbreviations, map it to the appropriate type, and put it at the end of the name.

The expected types for the street name in Cairo, Egypt are [street, square, road].

The expected abbreviations are [St,St. , Rd., Rd, Sq, Sq.]

1. Detecting street type abbreviations

I began detecting all names that have any of the expected abbreviations in any position of it and here is a sample:

'Rd': set(['Al Mansoureya Canal Rd’,'Al Masanea Rd','Al Nasr Rd’]),

'Sq.': set(['Othman ibn Affan Sq.', 'Victoria Sq.']),

'St': set(['13 St’,'21 St’]),

'St.': set(['2 Soliman Abaza St.Mohandeseen, Giza','51 Khedr El Touny St.']),

'st.': set(['Dr. Lashin st.','Future st.','Hamad Yassin st.’])

2. Mapping abbreviations to the appropriate name

Here we map the abbreviations to the appropriate name using the following mapping

```
mapping = { "St": "Street ","St.": "Street ","Sq":"Square",
            "Sq.":"Square","Rd":"Road","Rd.":"Road","st.":"Street"
          }
```

3. Correct street type position

The last step is to put the mapped types at the end of the street name if it exists at the beginning

St. Mark Cathedral => Mark Cathedral Street
Tunis St. => Tunis Street
Saint Mary and St. Marckos church => Saint Mary and Street Marckos church

2. Auditing postal codes

Although this data is only about Cairo it contains data about others city like 6th October, Fifth Settlement, First Settlement, 10th Ramadan

The reason for this may be because these cities were considered as part of Cairo and recently were formally separated.

The reason to mention this issue is related to the range and format of the postal code in this file.

The post codes are a 5 digits number and the range in Cairo is 11311 to 11668, but due to the issue I mentioned above we should include the range of the cities above in our valid range list.

So the valid ranges are:

Cairo 11311 to 11668
5th Settlement 11835
First Settlement 11865
10th Ramadan 44629, 44635, 44637
6th October 12566

Auditing Postal codes shows the following problems:

2.1 Inconsistent codes

[01066047247, 2500,31] which are not well formatted

2.2 Out of range codes

[11111,11231,12561]

These issues can not be handled by code as we can not replace the invalid post code to the correct value, So we can simply ignore this error or ignore the node or way within which this faulty post code exists.

After cleaning the street names for nodes and ways, ways names, and ways english name as shown above, I converted the cleaned OSM file to JSON array to be ready for mongodb.

3. Data overview with Mongoldb

1. Number of Documents

   ```
   db.cairo_egypt_map.find().count()
   ```

   ```
   365076
   ```

2. Number of nodes

   ```
   db.cairo_egypt_map.find({“type”:"node"}).count()
   ```

   ```
   320874
   ```

3. Number of ways

   ```
   db.cairo_egypt_map.find({“type":"way"}).count()
   ```

   ```
   44193
   ```

4. Number of unique users

   ```
   db.cairo_egypt_map.distinct(“created.user").length
   ```

   ```
   515
   ```

5. Top contributing user

   ```
   db.cairo_egypt_map.aggregate([
           {$group:{_id:"$created.user",count : { $sum : 1 }}},
           {$sort:{count:-1}},
           {$limit:5}
   ])
   ```

   ```
   { "_id" : "Allegro34", "count" : 37956 }
   ```

4. Additional Ideas

1. Top 10 amenities

```
db.cairo_egypt_map.aggregate([
                {$match:{amenity:{$exists:1}}},
                {$group:{_id:"$amenity",count : { $sum : 1 }}},
                {$sort:{count:-1}},
                {$limit:10}
        ])
```

```
{ "_id" : "place_of_worship", "count" : 390 }
{ "_id" : "parking", "count" : 322 }
{ "_id" : "restaurant", "count" : 196 }
{ "_id" : "school", "count" : 173 }
{ "_id" : "cafe", "count" : 147 }
{ "_id" : "fuel", "count" : 139 }
{ "_id" : "hospital", "count" : 119 }
{ "_id" : "bank", "count" : 106 }
{ "_id" : "fast_food", "count" : 101 }
{ "_id" : "pharmacy", "count" : 84 }
```

2. Top 5 cuisines

```
db.cairo_egypt_map.aggregate([
        {$match:{amenity:{$exists:1},
                cuisine:{$exists:1},”amenity”:"restaurant"}},
        {$group:{_id:"$cuisine",count : { $sum : 1 }}},
        {$sort:{count:-1}},
        {$limit:5}
])
```

```
{ "_id" : "regional", "count" : 19 }
{ "_id" : "italian", "count" : 8 }
{ "_id" : "pizza", "count" : 7 }
{ "_id" : "chicken", "count" : 7 }
{ "_id" : "burger", "count" : 6 }
```

3. Top 3 ways

```
db.cairo_egypt_map.aggregate([
        {$match:{“type":"way",name:{$exists:1}}},
        {$group:{_id:"$name",count : { $sum : 1 }}},
        {$sort:{count:-1}},
        {$limit:3}
])
```

```
{ "_id" : "الطريق الدائري", "count" : 93 }
{ "_id" : "طريق القاهرة - الاسكندرية الزراعى", "count" : 60 }
{ "_id" : "كورنيش النيل", "count" : 48 }
```

4. Top 5 cafes

```
db.cairo_egypt_map.aggregate([
        {$match:{amenity:{$exists:1},name:{$exists:1},”amenity":"cafe"}},
        {$group:{_id:"$name",count : { $sum : 1 }}},
        {$sort:{count:-1}},
        {$limit:5}
])
```

```
{ "_id" : "Cilantro", "count" : 9 }
{ "_id" : "Costa Coffee", "count" : 5 }
{ "_id" : "سيلانترو", "count" : 4 }
{ "_id" : "La Poire", "count" : 3 }
{ "_id" : "Coffeeshop Company", "count" : 2 }
```

This result shows the problem of writing the same place with different spelling or language, here we notice that the top cafe is Cilantro, and the third one is سيلانترو which is the same place as Cilantro but written in arabic. So here the actual count for Cilantro or سيلانترو is 13

5. Conclusion

It was interesting to work in this GPS data that are exist due to users’ contributions, it is really big and include a very useful information. Of course it has many problems like the listed above but I think it has been cleaned for the project’s purpose.

For the future I think that there is a need for a monitoring system to monitor the users’ inputs and put more standards for writing on OpenStreetMap.org to minimise the amount of uncleaned data.